International procedure specification for Logistics Support Analysis LSA

Issue 1  2010-04-01
Copyright and user agreement

1 Copyright
Copyright (C) 2010 AeroSpace and Defence Industries Association of Europe - ASD.
All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as may be expressly permitted by the Copyright Act or in writing by the Publisher.
S3000L™ is a trade mark owned by ASD.
All correspondence and queries should be directed to:
ASD
Monsanto Building
270 Avenue de Tervuren
B-1150 Brussels
Belgium

2 Agreement for use of the specification S3000L™ suite of information

2.1 Definitions
S3000L™ suite of information means, but is not limited to:
− the International procedure specification for Logistics Support Analysis LSA - S3000L
− examples (e.g. XML instances, pdf files, style sheets) and schemas
− any other software or information under the heading “S3000L™ suite of information”, available for download from www.asd-stan.org

Copyright Holder means AeroSpace and Defence Industries Association of Europe (ASD).

2.2 Notice to user
By using all or any portion of S3000L™ suite of information you accept the terms and conditions of this User Agreement.
This User Agreement is enforceable against you and any legal entity that has obtained S3000L™ suite of information or any portion thereof and on whose behalf it is used.

2.3 License to use
As long as you comply with the terms of this User Agreement, the Copyright Holders grant to you a non-exclusive license to use S3000L™ suite of information.

2.4 Intellectual property rights
S3000L™ suite of information is the intellectual property of and is owned by the Copyright Holder. Except as expressly stated herein, this User Agreement does not grant you any intellectual property right in the S3000L™ suite of information and all rights not expressly granted are reserved by the Copyright Holder.

2.5 No modifications
You must not modify, adapt or translate, in whole or in part, S3000L™ suite of information. You may however add business rules.
2.6 No warranty
S3000L™ suite of information is being delivered to you "as is". The Copyright Holder does not warrant the performance or result you may obtain by using S3000L™ suite of information. The Copyright Holder makes no warranties, representations or indemnities, express or implied, whether by statute, common law, custom, usage or otherwise as to any matter including without limitation merchantability, integration, satisfactory quality, fitness for any particular purpose, or non-infringement of third parties rights.

2.7 Limitation of liability
In no event will the Copyright Holder be liable to you for any damages, claims or costs whatsoever or any consequential, indirect or incidental damages, or any lost profits or lost savings or for any claim by a third party, even if the Copyright Holder has been advised of the possibility of such damages, claims, costs, lost profits or lost savings.

2.8 Indemnity
You agree to defend, indemnify, and hold harmless the Copyright Holder and its parents and affiliates and all of their employees, agents, directors, officers, proprietors, partners, representatives, shareholders, servants, attorneys, predecessors, successors, assigns, and those who have worked on the preparation, publication or distribution of the S3000L™ suite of information from and against any and all claims, proceedings, damages, injuries, liabilities, losses, costs, and expenses (including reasonable attorneys' fees and litigation expenses), relating to or arising from your use of the S3000L™ suite of information or any breach by you of this User Agreement.

2.9 Governing law and arbitration
This User Agreement will be governed by and construed in accordance with the laws of the Kingdom of Belgium.

In the event of any dispute, controversy or claim arising out of or in connection with this User Agreement, or the breach, termination or invalidity thereof, the parties agree to submit the matter to settlement proceedings under the ICC (International Chamber of Commerce) ADR Rules. If the dispute has not been settled pursuant to the said Rules within 45 days following the filing of a Request for ADR or within such other period as the parties may agree in writing, such dispute shall be finally settled under the Rules of Arbitration of the International Chamber of Commerce by three arbitrators appointed in accordance with the said Rules of Arbitration. All related proceedings should be at the place of the ICC in Paris, France.

The language to be used in the arbitral proceedings shall be English.
# Table of contents

The listed documents are included in Issue 1, dated 2010-04-01, of this publication.

<table>
<thead>
<tr>
<th>Document title</th>
<th>Chapter</th>
<th>Applicable to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to the specification</td>
<td>Chap 1</td>
<td>All</td>
</tr>
<tr>
<td>Purpose</td>
<td>Chap 1.1</td>
<td>All</td>
</tr>
<tr>
<td>Scope</td>
<td>Chap 1.2</td>
<td>All</td>
</tr>
<tr>
<td>How to use the specification</td>
<td>Chap 1.3</td>
<td>All</td>
</tr>
<tr>
<td>Maintenance of the specification</td>
<td>Chap 1.4</td>
<td>All</td>
</tr>
<tr>
<td>General requirements</td>
<td>Chap 2</td>
<td>All</td>
</tr>
<tr>
<td>LSA Business process</td>
<td>Chap 3</td>
<td>All</td>
</tr>
<tr>
<td>Configuration management</td>
<td>Chap 4</td>
<td>All</td>
</tr>
<tr>
<td>Influence on design</td>
<td>Chap 5</td>
<td>All</td>
</tr>
<tr>
<td>Human factors analysis</td>
<td>Chap 6</td>
<td>All</td>
</tr>
<tr>
<td>LSA failure modes and effects analysis</td>
<td>Chap 7</td>
<td>All</td>
</tr>
<tr>
<td>Damage and special event analysis</td>
<td>Chap 8</td>
<td>All</td>
</tr>
<tr>
<td>Logistic related operations analysis</td>
<td>Chap 9</td>
<td>All</td>
</tr>
<tr>
<td>Scheduled maintenance analysis</td>
<td>Chap 10</td>
<td>All</td>
</tr>
<tr>
<td>Level of repair analysis</td>
<td>Chap 11</td>
<td>All</td>
</tr>
<tr>
<td>Maintenance task analysis</td>
<td>Chap 12</td>
<td>All</td>
</tr>
<tr>
<td>Software support analysis</td>
<td>Chap 13</td>
<td>All</td>
</tr>
<tr>
<td>Life cycle cost considerations</td>
<td>Chap 14</td>
<td>All</td>
</tr>
<tr>
<td>Obsolescence analysis</td>
<td>Chap 15</td>
<td>All</td>
</tr>
<tr>
<td>Inservice feedback</td>
<td>Chap 16</td>
<td>All</td>
</tr>
<tr>
<td>Disposal</td>
<td>Chap 17</td>
<td>All</td>
</tr>
<tr>
<td>Interrelation to other ASD specifications</td>
<td>Chap 18</td>
<td>All</td>
</tr>
<tr>
<td>Data elements</td>
<td>Chap 19</td>
<td>All</td>
</tr>
<tr>
<td>Data exchange</td>
<td>Chap 20</td>
<td>All</td>
</tr>
<tr>
<td>Terms, abbreviations and acronyms</td>
<td>Chap 21</td>
<td>All</td>
</tr>
<tr>
<td>Terms, abbreviations and acronyms - Introduction</td>
<td>Chap 21.1</td>
<td>All</td>
</tr>
<tr>
<td>Terms, abbreviations and acronyms - Glossary of terms</td>
<td>Chap 21.2</td>
<td>All</td>
</tr>
<tr>
<td>Terms, abbreviations and acronyms - Abbreviations and acronyms</td>
<td>Chap 21.3</td>
<td>All</td>
</tr>
<tr>
<td>Data element list</td>
<td>Chap 22</td>
<td>All</td>
</tr>
</tbody>
</table>

End of data module
# Chapter 1

*Introduction to the specification*

## Table of content

<table>
<thead>
<tr>
<th>Chap</th>
<th>Description</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 1</td>
<td>Introduction to the specification</td>
<td>S3000L-A-01-00-0000-00A-040A-A</td>
</tr>
<tr>
<td>Chap 1.1</td>
<td>Purpose</td>
<td>S3000L-A-01-01-0000-00A-040A-A</td>
</tr>
<tr>
<td>Chap 1.2</td>
<td>Scope</td>
<td>S3000L-A-01-02-0000-00A-040A-A</td>
</tr>
<tr>
<td>Chap 1.3</td>
<td>How to use the specification</td>
<td>S3000L-A-01-03-0000-00A-040A-A</td>
</tr>
<tr>
<td>Chap 1.4</td>
<td>Maintenance of the specification</td>
<td>S3000L-A-01-04-0000-00A-040A-A</td>
</tr>
</tbody>
</table>

End of data module
Chapter 1.1

Purpose

Table of contents

| Purpose ......................................................................................................................................... | 1 |
| References ..................................................................................................................................... | 1 |
| 1 General...................................................................................................................................... | 2 |
| 2 Purpose ................................................................................................................................... | 2 |
| 3 Background.............................................................................................................................. | 2 |

List of tables

| References ...................................................................................................................................... | 1 |

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF-STAN-00-60</td>
<td>Integrated Logistic Support - UK MoD</td>
</tr>
<tr>
<td>DEX1AD</td>
<td>DEX1A&amp;D - Aerospace and defense business DEX for exchange of product breakdown for support</td>
</tr>
<tr>
<td>DEX3AD</td>
<td>DEX3A&amp;D - Aerospace and defense business DEX for exchange of a task specification</td>
</tr>
<tr>
<td>GEIA-STD-0007</td>
<td>Logistics Product Data</td>
</tr>
<tr>
<td>ISO 10303-239 PLCS</td>
<td>Product Life Cycle Support (PLCS)</td>
</tr>
<tr>
<td>MIL-HDBK-502</td>
<td>Department of defense handbook acquisition logistics</td>
</tr>
<tr>
<td>MIL-STD-1388-1A</td>
<td>Logistics support analysis - DoD</td>
</tr>
<tr>
<td>MIL-STD-1388-2B</td>
<td>DoD requirements for a logistic support analysis record</td>
</tr>
<tr>
<td>S1000D</td>
<td>International specification for technical publications using a common source database</td>
</tr>
<tr>
<td>S2000M</td>
<td>S2000M - International specification for materiel management</td>
</tr>
<tr>
<td>S4000M</td>
<td>International procedure handbook for the development of scheduled maintenance programs for military aircrafts</td>
</tr>
<tr>
<td>S5000F</td>
<td>Specification for operational and maintenance data feedback</td>
</tr>
</tbody>
</table>
1 General
This chapter gives a basic overview of the S3000L purpose including the history of the development.

2 Purpose
The Logistic Support Analysis (LSA) is one of the most important processes of product support. It is the principal tool to
- design the products relevant to maintainability, reliability, testability and to optimize life cycle cost
- define all required resources to support the product in its intended use, during in-service operation

S3000L defines the processes, general requirements and related information exchange governing the performance of LSA during the life cycle of aerospace and defense products. This specification may also be used for complex technical products from other industrial domains.

3 Background
The creation of this specification originated within the Aerospace and Defence Industries Association of Europe (ASD) in 2005. At that time, MIL-STD 1388-1A had already been cancelled and DEF STAN 00-60 had become too unique to UK-MoD acquisitions to be applicable as an international handbook for general purposes.

The lack of a common valid procedure resulted in the development of various program specific LSA handbook versions and associate IT-solutions. For each new program (such as Eurofighter, NH-90, Tiger, A400M and Gripen) these specific tasks had to be accomplished. This caused considerable effort for both industry and its military customers.

This situation prompted an initiative by ASD and the Aerospace Industry Association of America (AIA) to consider the joint development of a new common international specification for LSA.

This initiative started with an "Inaugural meeting on S3000L" held in Brussels on January 18th, 2006. The attendees gave their expectations on the need for an international LSA specification based on shortfalls in current standards, specifications and handbooks. It was agreed that there is not a shortfall of standards, there are too many of them.

In a subsequent project definition phase meeting in Munich in March 2006, the following basic requirements were identified:

The LSA specification shall
- generally be based on the processes of MIL-STD-1388-1A, MIL-HDBK-502 and DEF-STAN-00-60 and the activity model given by ISO 10303-239 PLCS
- be the handbook for creation and development of LSA data exchanged by
  - DEX1A&D - Aerospace and defense business DEX for exchange of product breakdown for support
  - DEX3A&D - Aerospace and defense business DEX for exchange of a task specification
  - another exchange mechanism
  with the intention to address the relevant data from the MIL-STD-1388-2B.
- use experience gained from the performance of LSA for the current programs such as Eurofighter, NH-90, Tiger, A400M, Rafale, Gripen, JSF
- include process application guidelines and rules for information exchange
- be tailorable and include guidelines for tailoring
- take into account current ISO/EN baseline documents
- enable interfaces to the suite of the ASD specifications S1000D, S2000M, S4000M and the S5000F (the latter in preparation)

The development work was then allocated to an international team of experts working under the joint chairmanship of AIA and ASD representatives. The following companies/organizations contributed to the work:

- AgustaWestland  UK
- Airbus Deutschland GmbH  Germany
- Boeing  United States
- Dassault Aviation  France
- EADS Casa  Spain
- EADS Military Air Systems  Germany
- Eurocopter  France
- LOGSA  United States
- MBDA  France
- Saab AB  Sweden
Chapter 1.2

Scope

Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>1</td>
</tr>
</tbody>
</table>

List of tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>References</td>
<td>1</td>
</tr>
</tbody>
</table>

List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig 1</td>
<td>2</td>
</tr>
</tbody>
</table>

References

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1 References</td>
<td>2</td>
</tr>
</tbody>
</table>

1 Scope

The specification S3000L is designed to cover all processes and requirements governing the performance of LSA.

- It provides rules for the usage of the product breakdown for logistic purposes and for the selection of LSA candidate items.
- It describes type and methodology for performance of the specified analyses.
- It gives guidelines on how to process the results of the analysis tasks and on how to achieve a cost-efficient support solution.
- It covers the interface between LSA and the support engineering areas (eg reliability, maintainability and testability).
- It covers the interface between LSA and the ILS functional areas as shown in Fig 1 (eg supply support, technical data services, special tools/test equipment or training).

The specification in particular describes the interface between industry (contractor) and the customer, which, when based upon contractual agreements, will provide the typical deliverables of LSA as given in this specification. Examples for typical deliverables of an LSA process are:

- reliable and maintainable products because of influence on design from a logistics perspective
- cost efficient support system
- logistics product data
- ILS support products
Activities which are described in detail within other ASD specification like the preparation of technical documentation (covered by S1000D) or the establishment of a materiel management process (covered by S2000M) are out of scope of S3000L. The same holds for prescribed methodologies how to perform technical/logistic analysis activities (e.g., performance of scheduled maintenance analysis corresponding to S4000M or detailed performance of level of repair analysis).
Chapter 1.3

How to use the specification

Table of contents

How to use the specification ........................................................................................................... 1
References ..................................................................................................................................... 1

1 General ...................................................................................................................................... 3
2 Application ................................................................................................................................ 3
2.1 S3000L Application policy ............................................................................................... 3
2.2 S3000L Application .......................................................................................................... 3
2.3 Tailoring of S3000L processes ........................................................................................ 3
3 Basic definitions .................................................................................................................... 3
4 Acronyms ................................................................................................................................. 3
5 Organization of the specification ............................................................................................ 4
5.1 Chapter 1 - Introduction to the specification ................................................................. 4
5.2 Chapter 2 - General requirements ................................................................................... 4
5.3 Chapter 3 - LSA Business process ................................................................................... 4
5.4 Chapter 4 - Configuration management .......................................................................... 5
5.5 Chapter 5 - Influence on design ....................................................................................... 5
5.6 Chapter 6 - Human factors analysis ................................................................................ 5
5.7 Chapter 7 - LSA failure modes and effects analysis ....................................................... 5
5.8 Chapter 8 - Damage and event analysis .......................................................................... 5
5.9 Chapter 9 - Logistic related operations analysis ............................................................. 5
5.10 Chapter 10 - Scheduled maintenance analysis ............................................................. 6
5.11 Chapter 11 - Level of repair analysis ............................................................................. 6
5.12 Chapter 12 - Maintenance task analysis ...................................................................... 6
5.13 Chapter 13 - Software support analysis ....................................................................... 6
5.14 Chapter 14 - Life cycle cost considerations ................................................................... 6
5.15 Chapter 15 - Obsolescence analysis ............................................................................. 6
5.16 Chapter 16 - Inservice feedback ................................................................................... 7
5.17 Chapter 17 - Disposal ................................................................................................. 7
5.18 Chapter 18 - Interrelation to other ASD specifications .................................................. 7
5.19 Chapter 19 - Data elements ......................................................................................... 7
5.20 Chapter 20 - Data exchange ....................................................................................... 7
5.21 Chapter 21 - Terms, abbreviations and acronyms ......................................................... 8
5.22 Chapter 22 - Data element list ..................................................................................... 8
6 Supporting publications ............................................................................................................. 8

List of tables

1 References ................................................................................................................................. 1

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 1</td>
<td>Introduction</td>
</tr>
</tbody>
</table>

Applicable to: All  
S3000L-A-01-03-0000-00A-040A-A  
Chap 1.3  
DMC-S3000L-A-01-03-0000-00A-040A-A_001_00_EN-US.doc  
2010-04-01 Page 1
<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 2</td>
<td>General requirements</td>
</tr>
<tr>
<td>Chap 3</td>
<td>LSA Business process</td>
</tr>
<tr>
<td>Chap 4</td>
<td>Configuration management</td>
</tr>
<tr>
<td>Chap 5</td>
<td>Influence on design</td>
</tr>
<tr>
<td>Chap 6</td>
<td>Human factors analysis</td>
</tr>
<tr>
<td>Chap 7</td>
<td>LSA failure modes and effects analysis</td>
</tr>
<tr>
<td>Chap 8</td>
<td>Damage and event analysis</td>
</tr>
<tr>
<td>Chap 9</td>
<td>Logistic related operations analysis</td>
</tr>
<tr>
<td>Chap 10</td>
<td>Scheduled maintenance analysis</td>
</tr>
<tr>
<td>Chap 11</td>
<td>Level of repair analysis</td>
</tr>
<tr>
<td>Chap 12</td>
<td>Maintenance task analysis</td>
</tr>
<tr>
<td>Chap 13</td>
<td>Software support analysis</td>
</tr>
<tr>
<td>Chap 14</td>
<td>Life cycle cost considerations</td>
</tr>
<tr>
<td>Chap 15</td>
<td>Obsolescence analysis</td>
</tr>
<tr>
<td>Chap 16</td>
<td>Inservice feedback</td>
</tr>
<tr>
<td>Chap 17</td>
<td>Disposal</td>
</tr>
<tr>
<td>Chap 18</td>
<td>Interrelation to other ASD specifications</td>
</tr>
<tr>
<td>Chap 19</td>
<td>Data elements</td>
</tr>
<tr>
<td>Chap 20</td>
<td>Data exchange</td>
</tr>
<tr>
<td>Chap 21</td>
<td>Terms, abbreviations and acronyms</td>
</tr>
<tr>
<td>Chap 22</td>
<td>Data element list</td>
</tr>
<tr>
<td>DEX1AD</td>
<td>DEX1A&amp;D - Aerospace and defense business DEX for exchange of product breakdown for support</td>
</tr>
<tr>
<td>DEX3AD</td>
<td>DEX3A&amp;D - Aerospace and defense business DEX for exchange of a task specification</td>
</tr>
<tr>
<td>S1000D</td>
<td>International specification for technical publications using a common source database</td>
</tr>
<tr>
<td>S1003X</td>
<td>S1000D and S3000L interface specification</td>
</tr>
<tr>
<td>S2000M</td>
<td>International specification for materiel management</td>
</tr>
<tr>
<td>S4000M</td>
<td>International procedure handbook for the development of scheduled maintenance programs for military aircrafts</td>
</tr>
<tr>
<td>S5000F</td>
<td>International specification for operational and maintenance data feedback</td>
</tr>
<tr>
<td>ISO 10303-239 PLCS</td>
<td>Product Life Cycle Support (PLCS)</td>
</tr>
</tbody>
</table>
1 General
This chapter gives an overview of
− how to apply S3000L to a project
− organization of the specification
and the fundamental reading rules.

2 Application
2.1 S3000L Application policy
It is intended that S3000L shall be the common LSA specification to be used by customers and contractors, eg governments, procurement authorities, support agencies, and industry. By agreement between customer and contractor, S3000L can be supplemented by additional international or national requirements for specific projects. The use of the specification and any supplemental processes should always be a subject of the contractual agreement between the customer and contractor. It is also the intention of industry that the specification shall be used, whenever possible, in projects involving other customers throughout the world.

2.2 S3000L Application
At the start of any project in which the S3000L procedures are to be used, it is necessary for the customer and contractor to agree how S3000L should be utilized and to jointly define the variables and options which S3000L provides. These agreements are recorded in the document commonly known as the project’s Guidance Conference Document (GCD). The process used by the customer and contractor to establish the information to go into the GCD is known as the "LSA Guidance Conference", which is described in detail in Chap 3.

In addition, in order to supplement the GCD, the project should also define an interchange specification (data and other deliverables). Depending upon the complexity of the project, this may be a stand-alone specification, or integrated within the GCD.

2.3 Tailoring of S3000L processes
In order to ensure efficient application, S3000L has been designed to allow users to select functionality that is appropriate to their specific projects.

Individual chapters may be included or excluded. Tailoring can also be applied in regard to number and range of LSA candidate items, in regards to the number of analyses per candidate item and in regard to data to be used during an analysis.

3 Basic definitions
the Product Any platform, system or equipment (air, sea, land vehicle, equipment or facility, civil or military)
project The task to develop, maintain and dispose of the Product

4 Acronyms
Throughout the S3000L a set of common acronyms are used to aid understanding and to minimize duplication. These acronyms are only explained in this chapter. The same abbreviation is used for all tenses, the possessive case and singular and plural forms of a given word. A complete list of abbreviations, acronyms and definitions is given in Chap 21.

The common acronyms are:

AIA Aerospace Industries Association of America
ASD European Aerospace and Defense Industries
Organization of the specification

S3000L is organized into chapters. When subjects are common to more than one chapter, they have been assigned to an appendix which must be read in conjunction with the related chapter.

The nature of the project using S3000L will determine the range of deliverables that are required and hence the depth to which the S3000L procedures need to be employed (tailoring). Details are given in the specification.

5.1 Chapter 1 - Introduction to the specification

Chap 1 provides a summarized view on purpose, background, scope and application of S3000L. It also explains the rules regarding the maintenance of S3000L and how to access the specification.

5.2 Chapter 2 - General requirements

Chap 2 provides general information regarding the set-up and administration of an LSA program as part of an overall development/ILS program. It describes the interfaces, dependencies and time constraints in regard to the other ILS disciplines within a model scenario.

5.3 Chapter 3 - LSA Business process

Chap 3 is designed to provide a full, in-depth description of the complete LSA process. It starts with the establishment of product usage data by the customer and LSA significant product design and performance data by the contractor. These are joint inputs for the LSA Guidance conference. The LSA Guidance conference establishes an agreement on rules regarding type and depth of breakdown and on methodology of LSA candidate item selection and identification.

Chap 3 further defines the type, methodology and acceptance of the potential analysis tasks. It covers the customer involvement in the LSA business process and the performance of the LSA Review conference. This is the final step in accepting the LSA results and defining the support concept. It also starts the production of ILS elements to support operation.
5.4 Chapter 4 - Configuration management

Configuration management is the discipline that ensures the proper identification and version of the configuration. It controls changes, and records the change implementation status of the physical and functional characteristics of the structure, system/subsystem, and equipment.

Configuration management therefore is the means through which integrity and continuity of the design, systems engineering and supportability are recorded, communicated, and controlled. Configuration management efforts result in complete audit traceability of decisions and design modifications.

This chapter defines the details and methodology of how the principles of configuration management are applied to the LSA process. Refer to Chap 4.

5.5 Chapter 5 - Influence on design

Influence on design is the goal of an industrial activity generally known as supportability engineering. Supportability engineering influences those product characteristics which are vital for enabling the operation of the product according to performance and design requirements and to minimize Life Cycle Cost (LCC).

The product characteristics to be influenced are primarily Reliability, Maintainability and Testability. The tools for supportability engineering are early Reliability, Maintainability and Testability activities/programs and the performance of Logistic Support Analysis.

Supportability engineering acts as the enabler between Design and Development and Integrated Logistics Support (ILS).

This chapter describes how the methods of supportability engineering are applied and how the results will be linked to the LSA process. Refer to Chap 5.

5.6 Chapter 6 - Human factors analysis

Chap 6 describes the relationship and integration between human factors engineering and the support analysis process. Like all other supportability engineering functions, human factors provide source data that is used by the support analysis team to determine maintenance crew and support equipment requirements. This relationship begins in the design review process and continues through the development of the maintenance task analysis.

5.7 Chapter 7 - LSA failure modes and effects analysis

Chap 7 covers the methodology and decision logic that will be applied for the identification of corrective maintenance tasks to be applied to the product in case of a failure occurrence during normal use.

5.8 Chapter 8 - Damage and event analysis

Chap 8 covers the methodology for identifying and justifying those maintenance tasks which are appropriate in case of special events or damage found on the product during general or detailed inspections.

5.9 Chapter 9 - Logistic related operations analysis

Chap 9 covers the methodology for identifying and justifying logistic related operations tasks. These are tasks which can neither be assigned directly to the area of direct usage of a product system (eg documented in a user's guide) nor to the area of maintenance (eg documented in a maintenance handbook). However, they are accomplished in operations and maintenance facilities in support of the day to day operations (eg packaging, handling, storage, mooring)

These tasks can be very important for the proper usage of any product. There are many operational aspects (ease of operation, usability, flexibility of usage, mobility, etc) that are restricted by logistics related operations tasks.
5.10 Chapter 10 - Scheduled maintenance analysis
The identification of required scheduled or preventive maintenance is of vital importance for the operation of a complex Product. Technical, commercial, operational and environmental factors must be carefully considered. The preventive or scheduled maintenance, along with the corrective maintenance gives a complete view of what must be expected in terms of the effort for overall maintenance activities. Refer to Chap 10.

5.11 Chapter 11 - Level of repair analysis
Chap 11 describes the actions to be taken when performing a Level of Repair Analysis (LORA) as part of any acquisition program. The requirement for a LORA is usually located in logistic support plans or similar documents.

LORA is a stand-alone activity that establishes an optimised maintenance or support concept using technical, commercial, operational and environmental data. The LORA should be performed on each LSA candidate item to determine the most cost-effective method to restore each item to its full serviceable condition.

5.12 Chapter 12 - Maintenance task analysis
Chap 12 covers the methodology of how to analyse an identified maintenance task concerning its logistic requirements including spare parts and consumables, support equipment, personnel, facilities and task duration information. Additional information such as task criticality, training needs, pre- and post-conditions, safety and environmental requirements should also be considered.

5.13 Chapter 13 - Software support analysis
Chap 13 provides the logistic analyst with a guideline on how to handle the specific requirements concerning software in the environment of maintenance and operation of technical systems. Additionally, the interrelation between software and hardware will be defined clearly and an explanation will be given on how to integrate software aspects in the overall LSA process.

In modern technical Products, software aspects are of increasing significance. More and more Product functions are performed by complex software packages. For hardware components, concepts and processes are established to guarantee a proper supportability of the system during its entire life cycle. These concepts and processes are documented in the LSA process.

Software has the same requirement. Software and hardware are of equal importance for the proper function of a product. For this reason, an analysis methodology should be applied to develop an adequate software support concept, called Software Support Analysis (SSA).

5.14 Chapter 14 - Life cycle cost considerations
Chap 14 covers the actions to be taken by the LSA analyst with regard to life cycle cost. LCC is the total cost of ownership of a product including all the direct and indirect costs.

The objective of LCC is to choose the most cost effective approach from a series of alternatives to achieve the optimum cost of ownership for the life cycle of the product. LCC is an economic model over the life cycle, including the cost of operation, maintenance and disposal (Recurring Costs). These normally exceed all the other costs such as development (Non-Recurring Costs).

LCC is a very useful tool in the determination of design configuration alternatives, and operational and support concepts.

5.15 Chapter 15 - Obsolescence analysis
Chap 15 describes the relationship between obsolescence analysis and the support analysis process.
In the development process it is the goal of LSA to avoid/control the use of components and materiel that are likely to result in obsolescence in the early operational phase.

During the inservice phase, LSA will be involved in the analysis of obsolescence events and will contribute to the definition of economic alternatives with regard to the maintenance and support concept.

The results of the obsolescence analysis effort will be stored in the LSA data repository.

5.16 Chapter 16 - Inservice feedback

Chapter 16 covers the methodology of how to feed inservice data back into the LSA business process. This feedback is required to compare the actual product performance with the predicted performance values established during the LSA process. Amongst others, this comparison will enable industry to manage performance based contracts.

This subject will also be covered by a new ASD specification, ASD S5000F - Specification for operational and maintenance data feedback, which is under preparation.

5.17 Chapter 17 - Disposal

Chapter 17 deals with the disposal of products and product components, eg their disposal from active and/or passive operation. Although disposal of products normally occurs during the last phase of the life cycle, disposal has to be considered in the very early phases of every program planning to acquire a product.

Adequate disposal activities may include:
- destruction/neutralization of toxic substances
- enabling a sustainable development, by recycling materials
- demilitarization of defense systems, to avoid weapons proliferation

Disposal activities of the life cycle phases are outlined in this chapter. Appropriate data will be generated and be used during the LSA process.

5.18 Chapter 18 - Interrelation to other ASD specifications

Chapter 18 addresses the benefits of using the ASD suite of ILS Specifications in conjunction with this specification. The LSA relationships to the following specifications are addressed:
- S1000D - International specification for technical publications using a common source database.
  Note: Issue 1 of S3000L is synchronized to S1000D at Issue 4.0.
- S2000M - International specification for materiel management
- S4000M - International procedure handbook for the development of scheduled maintenance programs for military aircrafts
- S5000F - International specification for operational and maintenance data feedback

5.19 Chapter 19 - Data elements

Chapter 19 defines a coherent data model supporting the S3000L LSA process and its interaction with the related business processes. These processes can either be dependent upon data coming out from LSA or be business processes that provide input to the LSA process. Additionally, the chapter defines all data elements that are used in the S3000L data model. A matrix gives for each data element the related chapter in the specification and it describes the data group (class) to which the data element belongs.

5.20 Chapter 20 - Data exchange

Chapter 20 deals with the interaction between the LSA process and its related business processes can be realized using computer based data exchange. Chapter 20 defines a set of Data Exchange
Specifications (DEX) which supports the events when data is to be exchanged between various engineering disciplines, such as design engineering, maintainability engineering, reliability engineering and technical publications development.

A DEX is a standardized message, based on ISO 10303-239 Product Life Cycle Support (PLCS).

The PLCS compliant data exchange specifications that supports subsets of the S3000L data model are the aerospace and defense DEX1A&D - Aerospace and defense business DEX for exchange of product breakdown for support and the DEX3A&D - Aerospace and defense business DEX for exchange of a task specification. These DEX can be used to exchange task data between an S3000L application and an S1000D Common Source Data Base.

For more information on how DEX1A&D and DEX3A&D can be used to support the exchange of task data between S3000L and S1000D refer to S1003X - S1000D and S3000L interface specification. Refer to Para 6.

5.21 Chapter 21 - Terms, abbreviations and acronyms
Chap 21 provides a glossary of terms, abbreviations and acronyms used within S3000L.

5.22 Chapter 22 - Data element list
Chap 22 defines all the data elements that are used as attributes in the S3000L data model and in the S3000L data exchange specifications.

A data element list is organized alphabetically by the data element name, and contains:

- Data element name
- Data element data
- Data element definition, contains a textual definition and a list of valid values
- Class name, identifies Classes in the S3000L data model where the data element is used as an attribute
- Unit of Functionality, identifies in which section of Chap 19 where the Class is defined
- S3000L chapter references, indicating which business process in S3000L that creates (C), reads (R), or updates (U) the respective data element.

An attribute list defines all the data elements that are used to define the S3000L data types

6 Supporting publications

For implementation and data exchange the following specifications/files have been developed in sync with S3000L:

- DEX1A&D - Aerospace and defense business DEX for exchange of product breakdown for support
- DEX3A&D - Aerospace and defense business DEX for exchange of a task specification
- S1003X - S1000D and S3000L interface specification

1 Available as an extractable zip-file including a set of html files
2 Available as an extractable zip-file including a set of html files
3 Available as a pdf file
Chapter 1.4

Maintenance of the specification

Table of contents

Maintenance of the specification........................................................................................................1
References........................................................................................................................................1
1 Maintenance of the specification.................................................................................................1

List of tables

Table 1 References..........................................................................................................................1

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

1  Maintenance of the specification

The S3000L suite of information will be maintained and further enhanced by a S3000L Steering Committee which will comprise representatives from member nations of ASD and AIA.

Any proposals relating to changes to the specification are to be managed by change proposals.

The suite consists of the following four specifications/files:

- S3000L - International procedure specification for Logistics Support Analysis
- DEX1A&D - Aerospace and defense business DEX for exchange of product breakdown for support
- DEX3A&D - Aerospace and defense business DEX for exchange of a task specification
- S1003X - S1000D and S3000L interface specification
# Chapter 2

## General requirements

### Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General requirements</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>2 Integrated logistic support program</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Integrated logistic support program implementation</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Integrated logistic support program supplier participation</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Progress, status and management reporting</td>
<td>4</td>
</tr>
<tr>
<td>2.4 Integrated logistic support in the system engineering process</td>
<td>4</td>
</tr>
<tr>
<td>2.5 ILS management</td>
<td>4</td>
</tr>
<tr>
<td>2.5.1 ILS manager responsibilities</td>
<td>4</td>
</tr>
<tr>
<td>2.5.2 ILS management</td>
<td>5</td>
</tr>
<tr>
<td>2.5.3 Master ILS program milestone schedule</td>
<td>5</td>
</tr>
<tr>
<td>2.5.4 ILS organization</td>
<td>5</td>
</tr>
<tr>
<td>3 Logistics support analysis program</td>
<td>6</td>
</tr>
<tr>
<td>3.1 LSA implementation</td>
<td>6</td>
</tr>
<tr>
<td>3.1.1 LSA activities and performance</td>
<td>6</td>
</tr>
<tr>
<td>3.1.2 Develop an early LSA strategy</td>
<td>7</td>
</tr>
<tr>
<td>3.1.3 LSA program plan requirements</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Program management principles</td>
<td>9</td>
</tr>
<tr>
<td>3.2.1 Integrated logistics support management</td>
<td>9</td>
</tr>
<tr>
<td>3.2.2 Operating logistics organizational structures</td>
<td>9</td>
</tr>
<tr>
<td>3.2.3 Formal logistics organizations</td>
<td>9</td>
</tr>
<tr>
<td>3.3 Logistics management objectives and policies</td>
<td>10</td>
</tr>
<tr>
<td>3.3.1 Logistics management objectives</td>
<td>10</td>
</tr>
<tr>
<td>3.3.2 Logistics management policies</td>
<td>10</td>
</tr>
<tr>
<td>3.4 LSA program organization</td>
<td>10</td>
</tr>
<tr>
<td>3.5 LSA management responsibilities</td>
<td>11</td>
</tr>
<tr>
<td>3.5.1 The LSA manager</td>
<td>11</td>
</tr>
<tr>
<td>3.5.2 Program LSA managers/leads</td>
<td>12</td>
</tr>
<tr>
<td>3.5.3 Program technical staff (logistics engineers/analysts)</td>
<td>12</td>
</tr>
<tr>
<td>3.5.4 Supportability analysis integrated product team</td>
<td>12</td>
</tr>
<tr>
<td>3.6 Elements of a LSA program plan</td>
<td>12</td>
</tr>
<tr>
<td>3.6.1 General requirements</td>
<td>12</td>
</tr>
<tr>
<td>3.6.2 Functional requirements identification</td>
<td>13</td>
</tr>
<tr>
<td>3.6.3 Unique functional requirements</td>
<td>14</td>
</tr>
<tr>
<td>4 Product development organization</td>
<td>14</td>
</tr>
<tr>
<td>4.1 Integrated product teams</td>
<td>14</td>
</tr>
<tr>
<td>4.2 Multidiscipline teams</td>
<td>15</td>
</tr>
<tr>
<td>4.3 Single point accountability</td>
<td>15</td>
</tr>
<tr>
<td>4.4 Empowerment</td>
<td>15</td>
</tr>
<tr>
<td>4.5 Management plans</td>
<td>15</td>
</tr>
<tr>
<td>4.6 Clear product definition and interfaces</td>
<td>16</td>
</tr>
<tr>
<td>4.7 Disciplined processes</td>
<td>16</td>
</tr>
<tr>
<td>4.8 Effective communications</td>
<td>16</td>
</tr>
<tr>
<td>4.9 Performance metrics</td>
<td>16</td>
</tr>
</tbody>
</table>

Applicable to: All

S3000L-A-02-00-0000-00A-040A-A

Chap 2

DMC-S3000L-A-02-00-0000-00A-040A-A_001_00_EN-US.doc

2010-04-01 Page 1
1 General

1.1 Introduction

The Logistic Support Analysis (LSA) program is considered to be a subset of Integrated Logistic Support (ILS). ILS has overall responsibility for the development of technical information and the support environment that will be used to support a product throughout its intended life cycle. The different disciplines in the context of supportability (e.g., technical documentation, spare parts, support equipment, personnel and training) need to be harmonized. The main disciplines are:

- Design interface
- Supply support and provisioning
- Support and test equipment
- Technical data/technical documentation
- Personnel and manpower
- IT/Software support
- Facilities
- Scheduled maintenance/maintenance planning
- Packaging, Handling, Storage and Transportation (PHST)
- Training and training devices

The LSA program is the principal source of technical data for ILS planning and resource decision. LSA will be used:

- To link system design and ILS requirements to system readiness thresholds and to define detailed support element requirements.
- Throughout the acquisition cycle to assess and alter system design and to establish and update support element requirements.
- As an important source of design related data for determining and integrating all logistics support requirements, for analyzing alternative design, operational, and support concepts, and for conducting tradeoffs among design and the various elements of logistics support of technical data for ILS planning and resource decisions.

The logistic analysis activities results and support resource data will be stored in the LSA database to give all program elements access to a common data source for evaluating the supportability of design alternatives and objectives. The integrated product teams’ practices will
ensure the timely interaction of the LSA activities and data for all elements of the design process.

1.2 Objective
ILS supported by an embedded LSA process ensures there is an open exchange of information to and between the logistic disciplines and to the design. It pursues two thrusts simultaneously:

- **Design for support**
  
  A focus on designs that minimize operation, maintenance, training, support tasks, and life-cycle costs while optimizing operational readiness.

- **Design of support**
  
  The design, development, funding, test, and acquisition of all support resources needed to assure optimum performance and readiness of the system in its intended operational environment and mission profiles.

1.3 Scope
This chapter is directed to ILS managers and LSA managers for both the customer and the contractor. The understanding of how a LSA program needs to be integrated into the super ordinate ILS program is of crucial importance. The approach for implementing a LSA program varies across organizations and sometimes across projects within an organization. Regardless of the organization or project, there should be policies in place to establish the functional responsibilities and expectations of the LSA program.

2 Integrated logistic support program

2.1 Integrated logistic support program implementation
The LSA program is the central element of the contractor's logistics program and provides the framework for monitoring and controlling the orderly and systematic development and execution of the ILS program, including identification of necessary corrective activities, communications, and follow-up procedures. LSA provides the data required to define and develop logistic elements support resource requirements. Data generated by performance of LSA activities are documented in data records directly related to the logistic elements.

Through the integrated product development system engineering process, the LSA program has been established as the analytical tool for determination of supportability requirements for the logistic elements. Since design and supportability requirements are fed into the LSA database, a direct relationship has been established between logistic-related design parameters such as, reliability, maintainability, and availability to readiness resource requirements.

Evaluation of LSA output reports, coupled with design process feedback, assists the contractor in defining supportability drivers and requirements. Additionally, due to the iterative nature of LSA, an audit trail is formed which documents the process used to make supportability decisions.

LSA input data from the engineering specialties and supportability tasks are fed into the LSA database. Reports are provided to the logistic members of the integrated product organization and feedback occurs within the integrated product organization. Logistics elements are directly related to each other through the LSA process and their impact on other elements is readily available through output reports using the common LSA database. Using these outputs, the contractor can determine, for example, what effect selection of a particular item of support equipment will have on maintenance planning, training requirements, technical data, and supply support. This information permits logistic element resource planners to make timely and intelligent inputs into the design process, which in turn permits optimization of support at the lowest logistic Life Cycle Cost (LCC).
2.2 Integrated logistic support program supplier participation
Suppliers are an integral part of the logistics organization, especially main suppliers within the program need to be integrated in the ILS process. The contractor will impose as much of the ILS program on suppliers as is practical, consistent with the suppliers’ capability and experience. The contractor conducts ILS procurement reviews to evaluate candidate suppliers based on their logistics and supportability program capability. This effort, similar to a source selection evaluation, surveys suppliers to verify that each supplier is capable of providing the logistic support necessary to satisfy program requirements.

2.3 Progress, status and management reporting
ILS planning and control regularly reporting will provide ILS progress, status and management report needs. Such a management report can include among other things:

− Master ILS program milestone schedule
− Baseline schedule for each logistic element
− Current status of each logistic element compared to the Statement of Work (SOW)

The required reporting system is dependent on the project itself. Standardized solutions for such reporting requirements are often not available or not applicable to specific project needs.

2.4 Integrated logistic support in the system engineering process
Contractor objectives for integration of logistics and supportability into the systems engineering process is to ensure that the traditional design process disciplines, and reliability and maintainability engineering are employed in the design-to-support parameters for the program support system. This objective is part of the overall program management objective to:

− Achieve operational availability and readiness thresholds
− Achieve reliability and maintainability thresholds necessary to meet these objectives
− Assign appropriate priorities to logistics and supportability element requirements in system design tradeoffs
− Identify support and manpower drivers

Systems engineering is the application of scientific and engineering resources to:

− Transform the operational need into a system configuration which meets effectiveness standards
− Integrate related technical parameters and assure compatibility in a manner which optimizes the system and the system's design
− Integrate the efforts of all engineering disciplines and specialties into the integrated product development systems engineering process

Logistics and supportability:

− Flows down system requirements
− Develop logistic support requirements
− Develop the support system
− Interacts with and makes direct inputs to the integrated product organisation during trade studies
− Participates in design analyses
− Responds to customer requirements and expectations

2.5 ILS management
2.5.1 ILS manager responsibilities
The ILS manager with the direction, control and authority to:

− Plan, coordinate, develop, produce and deliver quality products on schedule and within cost
− Assure the development of each logistic element is driven by and results from the LSA
Plan, coordinate, and schedule the resources necessary to test and evaluate the system, support and test equipment, training equipment, containers, associated logistics data and planned logistics resources

Ensure supportability and supportability-related features are incorporated into the system, support system, and training system design

2.5.2 ILS management

The ILS management functions to monitor and control the execution of ILS planning, coordinate efforts, monitor schedules, assess performance, and ensure timeliness and accuracy of inputs. ILS management also evaluates compliance with applicable specifications, requirements, and guidelines. Program status should be maintained for individual areas of responsibility.

ILS management meetings will be scheduled to be held in conjunction with other major program meetings, eg design reviews. Regular logistics technical interchange meetings will be held in direct support of the ILS management.

2.5.3 Master ILS program milestone schedule

The ILS Planning and Control process provides the planning, scheduling and reporting system necessary for effective ILS management of the program. Generation, implementation and updating of the master ILS program milestone schedule can be established as a four step process.

Initiate specified and expected contract requirements. These documents provide key system development, testing and operational dates, locations and hardware requirements.

Major ILS milestones are identified and updated for use in ILS plans, schedules and status reports. This step requires a master ILS program milestone schedule for the purpose of integrating the overall ILS program, determining the relationship among logistic element tasks, and merging the ILS program into a master program schedule.

Logistic element plans and schedules provide the logistic elements leader with a planning and control system. Baseline logistic element schedules are linked to the master ILS program milestone schedule and are provided in an ILS program progress, status and management report (or reports).

Provides detailed logistic resource scheduling and status reporting for items such as, spares and repair parts, technical data, training equipment.

2.5.4 ILS organization

The ILS organization has to be established carefully. All responsibilities must be clear and transparent for all participants within the ILS and LSA processes (especially in programs where cooperation between several companies has to be established). The top level ILS manager is responsible for the proper establishment of such an organization. ILS managers at the several cooperation partners can support the overall ILS manager. The same holds for the LSA organization. It is recommended to establish a top level LSA manager who supports the ILS manager. LSA managers at the participating partner companies within a cooperation project have to report to the top level LSA manager. It must be pointed out that a proper ILS/LSA organization has to be established on both sides, customer and contractor. The example given in Fig 1 shows a simple schema of an ILS/LSA management structure.
3 Logistics support analysis program

LSA is a systematic iterative process that integrates system design and support system requirements and evaluates supportability requirements relative to achieving the specified system/component sustainability and availability requirements. The analyses documentation process is optimized by employing an electronic database. The logistics relevant information consists of the identification, optimization, and traceability of ILS resources (e.g., spares, manpower; personnel; training; support and test equipment, training and training equipment, facilities, PHST and technical documentation).

3.1 LSA implementation

Design reviews and technical information meetings should be conducted throughout the program. Host design reviews and conduct in-house design reviews to ensure that Design Program goals and objectives are being achieved. The cognizant LSA analysts and integrated product personnel will be aware of supportability and supportability related design issues through direct participation in design reviews and/or by receiving all design review meeting minutes. Supportability concerns involving equipment design that arises during meetings, or throughout the overall and continuous design process, will be brought to the attention of the cognizant integrated product engineers. All design related supportability issues will be documented, including status, as necessary to ensure that a timely and appropriate resolution is obtained. LSA reviews will be conducted in conjunction with scheduled design/quarterly reviews.

3.1.1 LSA activities and performance

The LSA process includes the application of selected quantitative methods (refer to Fig. 2) to aid in the:

- Initial determination and establishment of logistic criteria as an input to system design
- Evaluation of various design alternatives
- Identification and provisioning of logistic support elements
− Final assessment of the system support capability during use

![Diagram showing LSA process]

The LSA program is based on an integrated product definition concept and can be based on the following major elements:

− A LSA program plan that identifies all the required LSA activities that must be performed in order to influence design for supportability and determines the appropriate logistic resources
− Scheduling that identifies the timing of the LSA requirements. The LSA schedules may be based on program phased needs and established to be mutually beneficial and supportive of other program requirements
− Assignment of responsibilities for performance of LSA activities to the design, supportability, and ILS personnel skilled and qualified for the activity
− Effective management of a wide range of design, supportability, and ILS disciplines

3.1.2 Develop an early LSA strategy

Develop a proposed LSA program strategy to be performed early in the acquisition program. This strategy will identify the scope of the proposed supportability objectives for the system/equipment, and the qualitative analysis that will be performed to provide the best cost benefit for the acquisition program. The LSA requirements will be analyzed in order to establish a comprehensive LSA program, including quantitative, qualitative and test LSA requirements for the systems, subsystems, and components comprising the overall system. The resultant LSA program shall be a tailored cost effective program that accomplishes each customer requested procedure and provides a system that meets or exceeds each customer LSA requirement.
3.1.3 LSA program plan requirements

The LSA Program Plan (LSA PP) describes the strategy for developing the LSA activities during the engineering and manufacturing development phase of the complete program. It identifies and integrates the LSA activities, identifies management responsibilities and activities, and establishes the approach for accomplishing the analysis activities. This plan provides a roadmap of what LSA will be performed and how it will be accomplished. Furthermore, this plan provides an approach to LSA and a basis to measure progress throughout the various phases of the program. The LSA process is iterative and dynamic in nature; therefore this plan will be updated to reflect current program status and planned changes.

This LSA PP describes the LSA process and the management structure established to execute this process. It contains details as to how the LSA process will be accomplished to satisfy the intended requirements. LSA is an iterative process that usually continues for as long as the item under analysis is in continuous use. Once the item has been retired, the LSA data may be used as baseline comparison data for future supported items.

The LSA PP describes the management, organizations, and procedures required to accomplish program requirements. It shall include the following:

- Description, interrelationships, and schedule of procedures to be accomplished by each organizational element concerned in support of the LSA program
- Statement of requirements for LSA program participants (e.g., subcontractors, suppliers, partner companies, agencies)
- Identification of LSA program monitoring points
- Dissemination of LSA requirements to design and description of techniques to be employed to ensure that desired LSA is inherent in the product design
- Description of the planned approach to LSA activities, and specific methods and sources to be employed to satisfy qualitative supportability analysis and resolve design issues
- Description of the LSA database, Work Breakdown Structure (WBS), Breakdown element identification numbering system, and handling government furnished information
- Description of how problems will be identified, controlled, and reported
- Description of provisions for updating the plan

Planning and coordination of LSA is the responsibility of logistics engineering management. LSA managers provide technical guidance in conducting LSA and have special analytical skills required to accomplish LSA-peculiar activities. Procedures must ensure supportability requirements are identified as an integral part of systems engineering and design. Supportability design baselines identify:

- Quantified reliability and maintainability comparative analysis requirements, like:
  - Maintenance man-hours per operating hour
  - Mean Time Between Failure (MTBF)
  - Mean Time to Repair (MTTR)
- The maintenance concept including:
  - On equipment activities
  - Off equipment activities
- Field/fleet supportability improvements and status
- Lessons learned and status
- Support drivers
- New technology requirements

In general the LSA PP can be a part of an overall ILS management plan or ILS PP. For more complex projects it is recommended to have a separate document. Many aspects to be covered are also harmonized and agreed between customer and contractor within a LSA Guidance Conference (GC). In Chap 3 a proposal is given how to organize the huge amount of information with the help of two different documents:
3.2 Program management principles

The logistics team is responsible for development of the support system. This team is a subset of the integrated product team responsible for the design, development, manufacture, operations, and support of a specific product.

3.2.1 Integrated logistics support management

The maintenance of an interface with the LSA program manager (LSA manager) to provide continuity between program management direction and the development of the LSA results is of crucial importance. A smooth execution of the LSA program by maintaining proper interface between the LSA program manager/lead and other functional management should be ensured.

The ILS manager is responsible for providing logistics and supportability expertise and resources to the program teams. In the product development process, each integrated product team leader controls the budget and is held accountable for development of that team’s products. Consequently, most logistics and supportability personnel are assigned to, and work with, the integrated product teams. Logistics and supportability resources should be provided to the integrated product teams to:

- Incorporate supportability into the design of the system and support system by providing reliability, maintainability, testability, human engineering, integrated diagnostics, and environmental suitability resources.
- Develop the support system and training system, and plan logistics resources by providing supply support, support equipment, technical data, facilities, PHST, manpower and personnel, and training and training equipment personnel.
- Accomplish logistics engineering in the product development systems engineering and design process.
- Provide for LSA, standardization, interchangeability and interoperability, and environmental assurance.

3.2.2 Operating logistics organizational structures

The operating organizational structure for a support program may be aligned with the WBS. The hierarchical characteristics of the WBS may be used to establish lines of responsibility, authority, accountability and reporting chains from the lowest level member to the program manager. Integration of logistics into the depicted organization is designed to:

- Reflect each logistics and supportability element identified by the WBS and system specification.
- Provide effective interfaces for logistics-related design functions most important to developing the support system, and achieving readiness goals.

Each ILS manager is vested with total responsibility for all aspects of that team’s products. Product team leaders at each organizational level combine the responsibilities of all subordinate product teams. This responsibility and reporting chain continues to the point that the program manager, as leader of the system team, has responsibility for all products. A significant feature of this organization is every product has a single point of responsibility, authority and accountability in the organization.

3.2.3 Formal logistics organizations

Functional leaders such as product support, production, and engineering department heads are members of the product team and provide support to the program manager. In that capacity, these leaders bring functional expertise and resources to assist in program execution and ensure uniform, timely assignment of needed skills to the product teams to assist in program execution and ensure uniform, timely assignment of added skill to the integrated product teams.
3.3 Logistics management objectives and policies

3.3.1 Logistics management objectives
Integration of logistics and supportability into the integrated product development program organization ensures:
- Design reflects test data assessment, supportability alternatives and tradeoff evaluations
- Detailed specification requirements
- Logistic resource planning is adjusted as necessary
- Operational availability and readiness thresholds are met
- The item is supportable in the expected operational environment
- The operational environments is/are accurately assessed
- The Support System achieves expected performance

An underlying logistic program objective is to identify and resolve supportability technical risk issues early, prior to beginning production and deployment of the product.

3.3.2 Logistics management policies
To achieve logistics management objectives, a proper organization should be established for design-for-supportability through the integration of design and the development of the support system and training system. This will be achieved by:
- Structuring an integrated product development organization which provides for active logistics participation and influence on design
- Structuring the ILS process to be continuously interactive, on a working level, with design engineering through the system engineering process
- Planning to work closely with customers and suppliers to develop the system

Additionally, established a LSA process which:
- Provides the logistic analysis procedures for integrating supportability requirements into the baseline design
- Requires the support system configuration match the system design configuration
- Provides the detailed maintenance planning and bottoms-up identification of total logistics resource requirements

As a mean of controlling, establish an ILS program progress, status and management reporting system which will document that program design, development, test and evaluation, and transition accomplishments meet or exceed logistics priorities and developing supportability requirements.

3.4 LSA program organization
The program manager is the final authority for effecting implementation of the LSA program integrated product development organization. LSA is part of the system engineering integration. LSA is an integral element in the integrated product team’s commitment of providing the highest quality products and services at the lowest possible cost.

The system engineering integration organization is responsible for defining the requirements analysis for the LSA program.
- Performing the functional analysis and allocation and providing them to the integrated product teams
- Providing product and process solutions which satisfies the supportability requirements
- Providing system analysis and control activities through the system engineering process

The integrated product team leader is responsible for completing the LSA program on his product.

The ILS manager is responsible for monitoring the LSA activities on all teams, ensuring commonality across the program and for delivery of LSA data submittals (if required).
program manager has final authority for all program functions. Supportability is an active member of the integrated product teams. The objective of these teams is to develop guidance in system design that meets performance, producibility and supportability requirements and achieves low LCC.

3.5 LSA management responsibilities
The following list should give an overview of the potential responsibilities of the different players in the LSA process.

3.5.1 The LSA manager
The typical responsibilities of a LSA manager can be, but not limited to the following:
- Implement established company operating policies and procedures concerning LSA.
- Accomplishes quality reviews of LSA data through in process reviews and formal reviews prior to data submittal.
- Addresses questions and coordinate required corrective actions pertaining to the development, implementation, and modification of integrated maintenance concepts and logistics resources and related problems/status.
- Assesses subsystem design and support concepts for supportability influence/impact for their assigned subsystems.
- Assists in resolution of problems pursuant to acquisition of supplier/customer-furnished LSA data necessary to support the assigned subsystems.
- Assists in the development and implementation of formal LSA review activities to ensure integration of maintenance concepts and compatibility with contractor and customer requirements for their respective system.
- Conducts LSA reviews with the customer and incorporate the results into the LSA database.
- Coordinates evaluation of design changes for LSA impacts and ensure impacts are highlighted to decision makers when design changes will have an adverse impact on ease and cost of support.
- Coordinates with appropriate Program and Functional management to ensure problems are resolved for their assigned subsystems.
- Documents LSA task analyses and coordinate LSA task scheduling and planning.
- Maintains technical liaison with design/supportability engineering and logistics groups.
- Manages the technical aspects of the LSA process and the documentation of design considerations and logistic resource identification.
- Monitors/coordinates the technical aspects of LSA relative to established program objectives, schedules, and directives.
- Participates in customer and supplier design and program reviews and ensures LSA is an agenda topic as appropriate.
- Participates in technical coordination meetings and design reviews with suppliers and/or the customer. Each review shall include formal review and assessment of supportability and related design requirements.
- Provides assistance and information to Program and Functional management in achieving contractor and customer objectives.
- Schedules LSA in accordance with engineering documentation release dates/support milestones, and accomplish LSA database management and configuration.
- Tracks LSA task accomplishment and participate in problem resolution.
- Establish company LSA operating policies and procedures.
- Help establish program statement of work and approve man-hour estimates for all LSA activity.
- Help assign personnel to programs and projects as required satisfying the LSA statement of work.
- Monitor and assist LSA program managers/leads in the performance of program statements of work.
− Establish training requirements for LSA personnel and LSA related tasks.

3.5.2 Program LSA managers/leads
The typical responsibilities of an overall program LSA manager (eg required within international projects with several participating companies) can be, but not limited to the following:

− Ensure LSA program requirements are being met
− Implement established company operating policies and procedures
− Control program cost and maintain program schedule
− Maintain a functional interface with team members, subcontractors and/or vendors
− Maintain a program interface with program management and the customer
− Be technically responsible for the accuracy of the LSA

3.5.3 Program technical staff (logistics engineers/analysts)
− Responsible for the performance of the data analysis
− Operate data collection software and generate reports
− Manage supplier and customer interfaces for data collection
− Have working relationships with engineering, all ILS elements, suppliers, and customer

3.5.4 Supportability analysis integrated product team
Participate in the development of an early LSA strategy

− Develop the LSA PP
− Participate in program and design reviews
− Coordinate the operational requirements
− Coordinate the requirements of mission hardware, software, and support system standardization
− Participate in developing the requirements for comparative analysis
− Coordinate the requirements for technological opportunities
− Coordinate the requirements of supportability and supportability related design factors
− Perform scheduled maintenance analysis
− Participate in the development of support system alternatives
− Participate in evaluation of alternatives and trade-off analyses
− Coordinate inputs from maintainability and packaging, and develop maintenance task analysis
− Coordinate the requirements for early fielding analysis
− Coordinate the requirements for post production support analysis
− Coordinate the requirements for supportability test, evaluation, and verification

3.6 Elements of a LSA program plan
3.6.1 General requirements
The LSA PP shall include the following elements of information, but not limited to, also refer to additional information concerning LSA PP and GCD in Chap 3), with the range and depth of information for each element tailored to the project phases:

− A description of how the LSA program will be conducted to meet the system and logistic requirements defined in the applicable project documents.
− A description of the management structure and authorities applicable to LSA. This includes the interrelationship between line, service, staff, and policy organizations.
− Identification of each LSA task that will be accomplished and how each will be performed
− A schedule with estimated start and completion points for each LSA program activity or task. Schedule relationships with other ILS program requirements and associated system engineering activities shall be identified.
A description of how LSA activities and data will interface with other ILS and system oriented tasks and data. This description will include analysis and data interfaces with the following programs, as applicable:

- Equipment design
- Equipment maintainability
- Equipment reliability
- Equipment testability
- Facilities/infrastructure
- Human factors integration
- Initial provisioning
- PHST
- Parts control
- Standardization
- Support and test equipment
- Survivability
- System safety
- Technical documentation
- Test and evaluation
- Training and training equipment

- Breakdown structure identification of items upon which LSA will be performed and documented, including software items. Identification of a LSA Candidate Item List (CIL), and LSA candidate item selection criteria. The list shall include all items recommended for analysis, items not recommended and the appropriate rationale for selection or non-selection.
- Explanation of the breakdown element numbering system to be used.
- The method by which supportability and supportability related design requirements are disseminated to designers and associated personnel.
- The method by which supportability and supportability related design requirements are disseminated to subcontractors and the controls levied under such circumstances.
- Government data to be furnished to the contractor.
- Procedures for updating and validating of LSA data to include configuration control procedures for LSA data.
- LSA requirements on Government Furnished Equipment/Materiel and subcontractor/vendor furnished materiel including end items of support equipment.
- The procedures to evaluate the status and control of each task and identification of the organizational unit with the authority and responsibility for executing each task.
- The procedures, methods and controls for identifying and recording design problems or deficiencies affecting supportability, corrective actions required, and the status of actions taken to resolve the problems.
- Description of the data collection system to be used by the contractor to document, disseminate and control LSA and related design data.

### 3.6.2 Functional requirements identification

This activity identifies the required operations and support functions for the product. The primary inputs are the results of the supportability and supportability-related design factors, and results of the reliability and maintainability programs and analyses. The results of this task form the basis for the support system alternatives. Through coordination and interaction with design engineering, maintainability, human engineering and technical manuals, LSA personnel will revise as applicable functional block diagrams identifying subsystem component changes and their functional interface within the subsystem and with associated subsystem components. LSA personnel will then evaluate and identify operations and maintenance functions that are required to implement the maintenance and operational concept for LSA candidates.
The functional requirements inherent in maintenance and operation of the product include requirements such as:
- Inspections
- Servicing
- Testing
- Operating
- Repairing
- Configuring for specific usage

3.6.3 Unique functional requirements
The same process used to identify operations and maintenance functions will identify any unique functional requirements. These unique functional requirements are normally associated with new technologies and equipment incorporated in the system or its support system. Hazardous materials, hazardous wastes, and environmental pollutants will also be considered.

Although the same analysis techniques are utilized to identify and evaluate these unique functions, special attention will be given to them because they have a tendency to exhibit higher risks. The LSA organization will ensure that configuration changes are tracked and their impact on identified unique functional requirements are identified and evaluated.

4 Product development organization
The product oriented organization integrates the unique capabilities of each engineering discipline through the product development process. This process is a systematic approach to the integrated, concurrent design of products and their related processes, including production and support. Use of this approach is intended to cause the developer, from the outset, to consider all elements of the product life cycle from conception through disposal, including cost, schedule, performance, supportability, quality and user requirements.

Integrating product development during design will help achieve first time quality and improved compliance with requirements will:
- Support completing the program within target cost
- Meet affordability requirements through design for produceability and concurrent manufacturing process improvements
- Reduce operational and support costs through design-to-supportability with emphasis on reliability, maintainability and life cycle cost

Fundamental characteristics of an effective integrated product development organization are:
- Integrated product teams
- Clear product definition and interfaces
- Multidiscipline teams (integrated teams)
- Disciplined processes
- Single point accountability
- Effective communications
- Empowerment
- Performance metrics

4.1 Integrated product teams
Each of the product items in the entire system, including supplier and customer furnished items, is owned by a product team. Resource allocation and integration normally occur through a WBS hierarchy of product teams starting with the system team. The flow down is from team to team, down to the detailed subassembly or component. These teams represent significant, identifiable subsystems, subassemblies and components of the system, operation planning and logistics segments.
Each level in the organization is fully accountable for the aggregate of its subordinate teams, as well as for the integration and overall performance of its end product. Team leaders at all levels embody the full responsibility, authority and accountability of their teams. A single unbroken line of authority flows through the team leaders from the program manager to the subassembly or component team leaders. Team leaders allocate roles and responsibilities based on specific, tangible products. Each team member has a clear charter that focuses on the development or creation of a product.

4.2 Multidiscipline teams

Early consideration of the product life cycle results in the identification of design conflicts. This feature is made possible with multidiscipline teams where there is diversity of perspective to a common mission. Each individual, whether they are skilled in

- Design
- Quality assurance
- Manufacturing
- Reliability, testability and maintainability
- Technical data and technical documentation
- Support equipment
- Supply support
- Facilities
- Technical services
- PHST
- Personnel and training

are forged into a single effort. The objective of an integrated team is to identify and resolve as much design conflict as possible in the shortest period of time with the understanding that each perspective is of equal value. Each team member brings their unique perspective to the creation of a single, concurrently engineered design. Without this integration of perspectives there is no integrated product development or concurrently engineered design.

4.3 Single point accountability

The integrated product team leader is held accountable for the development of all products. This feature provides the benefits of:

- avoiding redundant efforts

and

- defining clarity of purpose.

This individual brings together the right resources (manpower, skills, budget and facilities) to achieve the mission.

4.4 Empowerment

Empowerment enables integrated product team members to achieve their mission and provides them with the ability to influence and to the maximum extent possible, control their disciplines destiny. Individuals are empowered to the extent necessary to fulfill their individual roles and responsibilities but their ultimate destiny is controlled by the performance of their teammates. By definition, every individual within a team is dependent upon the other team members. Empowerment is the principle of enabling team members, with sufficient and adequate resources, to achieve their specified roles and responsibilities.

4.5 Management plans

Corrective actions for resource problems consist of procurement actions (early orders for long-lead-time items or schedule, cost, and budget changes), support system/concept changes, or design changes. These resource challenges will be brought to management's attention through
technical analysis reviews and schedules. Problems that are identified will be worked through the responsible functional management. The responsible subsystem logistics manager will ensure that appropriate action is assigned and that proposed changes are implemented.

4.6 **Clear product definition and interfaces**
Mission success is dependent upon understanding clearly and completely what the product is that the team is accountable for providing. The product description includes not only the requirements like performance, supportability, and produceability but also how that product must relate or interface with other products.

4.7 **Disciplined processes**
The processes for designing, developing, testing, producing and supporting system and support system products are founded upon accepted standardized principles. The focus of integrated product development is rigorous and consistent application of standardized processes for the development of all products along with a continuous search for methods to improve the process. As the processes are documented, the team is committed to management by process rather than management of the product.

4.8 **Effective communications**
Communications in the integrated product development process simply means that a free and open exchange of data, information and opinions is the fuel that keeps the process working. Effective communications begins with establishing the roles and responsibilities of the teams, defining the team’s products, establishing team goals and objectives, and providing the ability to identify and resolve design conflict. Instrumental to effective communications are collocation of team members, regularly scheduled integrated product team meetings, all-hands meetings, functional and program staff meetings, effective use of memos, use of electronic communications (eg E-mail, video conferences), and knowledgeable interchange of data and information.

4.9 **Performance metrics**
It is important to know how well each integrated product team is progressing toward development of their products. Measuring is achieved by identifying those critical parameters in the product development process and establishing an associated metric. The parameters selected by an individual or team must characterize:

- Product quality
- Schedule
- Cost
- Risk assessment

End of data module
# Chapter 3

## LSA Business process

### Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSA Business process</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>6</td>
</tr>
<tr>
<td>1 General</td>
<td>6</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>6</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>7</td>
</tr>
<tr>
<td>2 Establishment of product usage data</td>
<td>7</td>
</tr>
<tr>
<td>2.1 General usage aspects</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Operational requirements document</td>
<td>9</td>
</tr>
<tr>
<td>2.2.1 General usage scenario (overview)</td>
<td>9</td>
</tr>
<tr>
<td>2.2.2 Geographical position of locations and special conditions of each location</td>
<td>9</td>
</tr>
<tr>
<td>2.2.3 Systems (products) supported and systems (products) deployment</td>
<td>10</td>
</tr>
<tr>
<td>2.2.4 Usage overview</td>
<td>10</td>
</tr>
<tr>
<td>2.3 Detailed checklist for the creation of an operational requirements document</td>
<td>10</td>
</tr>
<tr>
<td>2.4 Customer requirements document</td>
<td>10</td>
</tr>
<tr>
<td>2.4.1 Supply concept</td>
<td>10</td>
</tr>
<tr>
<td>2.4.2 Support equipment concept</td>
<td>10</td>
</tr>
<tr>
<td>2.4.3 Personnel integration and staff training</td>
<td>11</td>
</tr>
<tr>
<td>2.4.4 Facilities</td>
<td>11</td>
</tr>
<tr>
<td>2.4.5 IT and communication resources</td>
<td>11</td>
</tr>
<tr>
<td>2.4.6 New organizational structures</td>
<td>11</td>
</tr>
<tr>
<td>2.4.7 Schedule considerations</td>
<td>11</td>
</tr>
<tr>
<td>2.4.8 Additional aspects</td>
<td>11</td>
</tr>
<tr>
<td>2.5 Detailed checklist for the creation of a customer requirements document</td>
<td>12</td>
</tr>
<tr>
<td>2.6 Time schedule for document creation</td>
<td>12</td>
</tr>
<tr>
<td>2.7 Site surveys</td>
<td>12</td>
</tr>
<tr>
<td>2.8 Qualification requirements</td>
<td>12</td>
</tr>
<tr>
<td>2.9 Certification requirements</td>
<td>12</td>
</tr>
<tr>
<td>3 Establishment of product design and performance data</td>
<td>13</td>
</tr>
<tr>
<td>3.1 Selection criteria concerning LSA relevant data and information</td>
<td>13</td>
</tr>
<tr>
<td>3.1.1 Selection of LSA data and information derived from contractual documents</td>
<td>13</td>
</tr>
<tr>
<td>3.1.2 Selection of LSA data and information derived from product usage data</td>
<td>13</td>
</tr>
<tr>
<td>3.1.3 Selection of LSA data and information derived from design and performance specifications</td>
<td>14</td>
</tr>
<tr>
<td>3.1.4 Selection of LSA data and information relevant for product certification and verification contained in other product documents</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Influence of overall LSA strategies and principles on LSA data selection</td>
<td>14</td>
</tr>
<tr>
<td>3.2.1 Procedures and principles influencing selection of LSA data</td>
<td>14</td>
</tr>
<tr>
<td>3.3 Acceptance rules concerning the verification of values</td>
<td>15</td>
</tr>
<tr>
<td>3.3.1 Category of measurement values</td>
<td>15</td>
</tr>
<tr>
<td>3.3.2 Identifying tolerances</td>
<td>15</td>
</tr>
<tr>
<td>3.3.3 Establishment of acceptance criteria</td>
<td>16</td>
</tr>
<tr>
<td>3.3.4 Establishment of special rules</td>
<td>16</td>
</tr>
<tr>
<td>3.4 Criteria and procedural aspects for verification of projected values</td>
<td>16</td>
</tr>
<tr>
<td>3.4.1 Determination of measurable target values</td>
<td>16</td>
</tr>
<tr>
<td>3.4.2 Verification of the individual projected values</td>
<td>16</td>
</tr>
<tr>
<td>3.4.3 Verification method</td>
<td>16</td>
</tr>
<tr>
<td>3.4.4 Verification for special purposes</td>
<td>16</td>
</tr>
</tbody>
</table>
List of tables

1  References ......................................................................................................................6
2  Required input documents for LSA GC .........................................................................21
List of output documents of the LSA GC ....................................................................... 22
Basic categories of breakdown elements ....................................................................... 28
Basic table for explicit breakdown documentation ........................................................... 39
BEI readability aspects .................................................................................................... 40
Product variant applicability from Fig 18 ....................................................................... 43
Aspect of replaceability .................................................................................................. 44
Aspects of repairability .................................................................................................. 44
Definitions of MSI and MRI ........................................................................................... 46
Definitions of Structural Item, Structure Significant Item and Structural Detail ................. 47
LSA candidate categories .............................................................................................. 47
Classification criteria for full LSA candidates .................................................................. 50
Classification criteria for partial LSA candidates ............................................................. 50
Classification criteria for LSA candidate family ............................................................ 51
Recommendations for LSA candidate selection ............................................................ 52
Depth of maintainability analysis depending of item type ............................................... 57
Classification of different depths of LORA .................................................................... 59
Analysis activities selection criteria ............................................................................... 63
Explanation of time schedule of the commenting process ............................................. 70
Examples for status codes ............................................................................................ 75
List of different spare part types identified by LSA ........................................................... 78
Checklist of detailed questions to support the ORD creation ........................................... 82
Checklist of detailed questions to support the CRD creation ........................................... 86
Listing of functional breakdown from Fig 28 ................................................................... 89
Listing of functional breakdown from Fig 29 ................................................................... 90
Listing of mixed breakdown extract from Fig 30. ........................................................... 91
Listing parent-child breakdown extract from Fig 33 ...................................................... 93

List of figures
1 Schedule for the creation of the basic documents .......................................................... 12
2 LSA tasks in conjunction with the preparation of an offer/contract ................................. 18
3 Flowchart of LSA business process at a glance (Sheet 1 of 2) ........................................ 19
4 Flowchart of LSA business process at a glance (sheet 2 of 2) ....................................... 20
5 LSA GC, inputs and outputs ....................................................................................... 21
6 Example of content of a LSA Program Plan ................................................................ 23
7 Example of content of a Guidance conference document ........................................... 25
8 Simple functional breakdown ..................................................................................... 30
9 Simple physical breakdown ......................................................................................... 30
10 Breakdown methodology - Simple parent-child philosophy ....................................... 31
11 Breakdown methodology - Extended parent-child philosophy ................................. 31
12 Breakdown methodology - Mixture of functional and physical methodology ............ 32
13 Example for a spare part grouping concept ................................................................ 33
14 Product breakdown - Traditional LSA approach versus PDM approach ...................... 34
15 Product breakdown derived from Table 5 .................................................................... 40
16 Example for simple BEI syntax (reduced to numbering system) .................................. 41
17 Example for extended BEI syntax ............................................................................ 41
18 Usage of BEI and applicability ............................................................................... 43
19 Analysis activities relations and overview .................................................................. 62

Applicable to: All
References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 5</td>
<td>Influence on design</td>
</tr>
<tr>
<td>Chap 9</td>
<td>Logistic related operations analysis</td>
</tr>
<tr>
<td>Chap 10</td>
<td>Scheduled maintenance analysis</td>
</tr>
<tr>
<td>Chap 12</td>
<td>Maintenance task analysis</td>
</tr>
<tr>
<td>Chap 13</td>
<td>Software support analysis</td>
</tr>
<tr>
<td>DEF-STAN-00-60</td>
<td>Integrated Logistic Support - UK MoD</td>
</tr>
<tr>
<td>MIL-STD 1388-2B</td>
<td>DoD requirements for a Logistic Support Analysis Record</td>
</tr>
<tr>
<td>S1000D</td>
<td>International specification for technical publications using a common source database</td>
</tr>
<tr>
<td>S4000M</td>
<td>International procedure handbook for the development of scheduled maintenance programs for military aircrafts</td>
</tr>
</tbody>
</table>

1 General

1.1 Introduction

With the introduction of a new product, all logistic requirements must be made available in a timely manner. This requires programs to establish a process to ensure consideration of logistic requirements during the design of the product. This process includes a number of analysis activities concerning a wide range of logistic considerations. Activities to achieve proper
supportability must be carefully agreed to between the contractor and the customer. Early consideration of logistic aspects is increasingly important with regard to both operational and economic considerations. A product that cannot be operated and maintained properly and cost effectively is not acceptable to the end user.

It should be noted that modern products normally contain software elements. The overall Logistic Support Analysis (LSA) process for software and hardware is very similar. For this reason, this chapter is valid for the supportability of software, as well as hardware. However, some specific aspects of software must not be ignored and will be covered by Chap 13, Software Support Analysis (SSA). This chapter is referenced as appropriate.

Logistic considerations are significant in terms of costs. Over the life cycle of a complex technical product, support costs are much higher than acquisition costs. Because of this, programs are tending to consider logistic aspects as important as performance aspects.

1.2 Objective
This chapter is a guideline for the establishment of an effective LSA process. It considers each life cycle phase of a technical product and emphasizes the importance of the logistic requirements. Additionally, the interaction between the contractor and the customer during the different phases of the product's life cycle is described in detail. The LSA process has two main purposes: 1) with the help of different analysis methods, the design should be influenced to allow appropriate supportability, and 2) the creation of logistic end products should be managed. The requirements for spare parts, consumables, technical documentation, support equipment, personnel, training, facilities and software support are identified and the creation of the logistic end products should be supported. All together, this is an extensive task, which has to be carried out carefully. A properly established LSA process that is embraced and used by the customer and the contractor is an example of an efficient plan for a successful introduction of new complex technical equipment.

1.3 Scope
This chapter is directed to Integrated Logistic Support (ILS) managers and LSA managers for both the customer and the contractor. LSA offers a powerful methodology to build the core needs for the realization of ILS. Additionally, monitoring and control functions can be achieved by using relevant LSA information. The quality and commonality of the logistic products can be positively influenced by the application of a LSA process. With the incorporation of various analysis results, it can be assured that customer's needs for operability, supportability and readiness can be achieved.

2 Establishment of product usage data
To identify the pertinent supportability aspects of a new product, all relevant information related to the intended use must be collected and documented carefully. This chapter describes how information can be structured and documented in a set of obligatory documents. It should be pointed out that in an early stage of a project sometimes complete relevant information is not always available. In this case, iterative steps may be necessary for a complete definition of the customer's usage of the product to be analyzed. In any case, changes to the usage scenario are of crucial importance for logistic purposes and must be taken into consideration by the logistic analysts accordingly.

2.1 General usage aspects
To establish an initial overview to identify pertinent supportability aspects, some very general decisions are necessary. These first decisions must consider the overall preconditions for the usage of the new product to be introduced and also consider some strategic aspects coming from the product design and performance data, which are described in Para 3. These general decisions should be documented in a general project document describing the overall support strategy. This strategy should be a basic guideline for the further execution of any logistic
analysis and for creation of the Operational Requirements Document (ORD) and the Customer Requirements Document (CRD).

The general questions, which should be answered before or at least in parallel to the creation of the ORD and the CRD, are the following:

- What kind of product should be used?
  A short description of the product should be given including key features, key requirements and basic technical data needs.

- Will contractor logistics support be required? If yes, at which maintenance levels?
  This is a very central question that can only be answered by a deeper examination of some additional aspects described in the following questions.

- Where will the product be operated and maintained?
  
  - In which operational environment will the product be operated and maintained (from a fixed/industrial/benign environment to a mobile/austere/hostile one)?
  
  - Will the environment influence the item’s characteristics (e.g., reliability, maintainability)?
  
  - Will the environment significantly change the manner in which the item must be repaired? If so, contractor support might not be the best approach.

- How long will the product be used (predicted service life)?
  If the product will only be in the inventory for a few years, then contractor support might be preferable to a lengthy and costly gearing-up of an organic logistics support structure.

- How many maintenance levels are planned for the product support?
  The repair strategy must be clearly envisioned before the ORD is written.

- What are the features and/or typical actions within each maintenance level?
  For example in a classical 2-level maintenance concept, the exchange of "black boxes" and the re-sending to the contractor or to the original manufacturer will be typical.

- Are there existing maintenance capabilities that can be adapted to the support concept for the new product?

- Is the usage of existing maintenance capabilities from former products or other similar products at or nearby the operating locations possible and effective?

- Should the customer be involved in any repair activity of the product or should the corrective maintenance be limited to simple "black box" exchange (replace only)? The involvement of the customer is dependent on the abilities which can be provided by the customer itself.

- Which frequency and duration of preventive or scheduled maintenance actions (expressed in measurable terms) is acceptable?

- Are there limitations for preventive maintenance e.g. because of special preconditions concerning available personnel or facilities?

- How much of the software is mature? How much is customer unique?
  Software, never delivered 100% bug-free, may take several years to mature. The logistics support structure should also address software maintenance of potential user required upgrades.

- Is there any software/data loading and/or unloading which should be performed by the customer?
  Which concept should be established to guarantee proper function of the product after the loading of software/data at an acceptable level (e.g., simulated integrity test based on a hardware-software compliance matrix in a test bench, General Purpose Test Equipment concept for software, required software device and encrypting system for loading and/or unloading).
What is the expected need for product replacement or upgrade due to changes in technology? These questions concern how a support structure can keep up with changes in the product and modify the support strategy. If it is difficult or impossible, contractor logistics support should be preferred.

All general questions should be answered as completely and carefully as possible. Additional questions to those listed above may occur. This depends on the project and on the corresponding products to be used. Best information available should be the basic principle. Create a basic document with all general and programmatic information and harmonize this document between the contractor and the customer. Ideally this should be done before the creation of the ORD and the CRD.

2.2 Operational requirements document

Operational requirements must be defined in a quantitative and qualitative way. Previously conducted analyses concerning area of operation and product usage, that identified relations between hardware, usage and supportability, should be considered. A further definition of the identified operational requirement must be performed by measurable details. Supportability requirements are developed directly from the operational requirements. Each supportability requirement should be based on an operational requirement, and that relationship should be clearly identified. If the basis for the supportability requirement is not clear, that requirement should be regarded with suspicion. The main goal is to identify and document the pertinent operational requirements related to the intended use of the new product. The writers of the ORD are required to identify key performance indicators. Any parameter not identified as a key parameter, should be a candidate for revision when supportability attributes are negatively impacted. The ORD requires the product developer to make difficult choices between “must have” and “nice to have” at an early stage of the program. This information is of vital importance to the logisticians so they can understand what they must support, regardless of costs, and what they can trade off.

Operational requirements to be considered within the ORD are described in the following paragraphs.

2.2.1 General usage scenario (overview)

In the general usage scenario, all information should be collected to describe the environmental aspects of the usage of the product.

- General description of overall usage areas and/or mission areas
- Possible operation scenarios and requirements for each scenario must be documented.
- Influence on the environment by the usage of the product and resulting actions to avoid or reduce negative influence
- Supportability problems that have arisen over the life of the currently used product
- Interaction and dependability with existing products
- Mobility requirements

2.2.2 Geographical position of locations and special conditions of each location

In this paragraph, details of each location should be collected to describe the special aspects of each location in which the product will be operated.

- Number and geographical position of the operating locations.
- Type of each operating location.
- Special conditions at each operating location.
- Is the location in a peacetime or wartime region?
  - Threat situations require a special emergency maintenance concept limited to minimum intervention.
- Is sufficient infrastructure available to access each operating location?
- Special infrastructural requirements for reaching a location.
− Capabilities (existing and planned) of each location (eg equipment, infrastructure, personnel, facilities, supply depot, repair station).
− Interactions between different locations concerning, as an example maintenance support, from one location for another location.

2.2.3 Systems (products) supported and systems (products) deployment
In this paragraph, the deployment of the products and the interactions coming from the deployment should be collected. The location at which the products will be based affects the decision to use either organic or contractor support.

− Number of systems supported per location
− Deployment of products per location
− Interactions between the different locations concerning usage

2.2.4 Usage overview
Depending on the customer, define the planned usage of the product as a basic input to logistic analysis. The frequency and the duration of the usage, in combination with the reliability of the product, provide the initial basis for determining the range and quantity of support resources that will be required.

− Key usage/key missions per location
− Performance parameters and constraints. The product performance parameters, such as range, accuracy, payload or speed, are to be identified in measurable quantifiable terms. General terms or those whose interpretation is potentially ambiguous must be avoided.
− Predicted operational availability and usage success rates
− Operation per unit of time
− Operation profile per operating day, week, month or year
− Usage of the product as training equipment as a portion of operating time
− Permanent operational conditions/possible maintenance windows
− Average duration of each unique usage event
− Key measurement base of usage per unit of time

2.3 Detailed checklist for the creation of an operational requirements document
For a detailed checklist on how to create an ORD, please refer to Para 11.1.

2.4 Customer requirements document
To ensure the proper usage of the system, the customer has to consider all the logistic requirements. The logistician can most efficiently influence the supportability and design of a new system during its early stages of development. One method to convey these needs is to document them in a CRD. The logistics elements of the CRD are described in the following paragraphs.

2.4.1 Supply concept
The supply concept is of crucial importance for the logisticians. A general decision on how to organize the provision of spare parts and consumables can have high impact on the maintenance scenario itself. It must be decided whether or not facilities must be planned because of the need for supply storage areas and who will manage the supply chain when outsourcing the Supply Chain Management. Costs (especially facility construction) and obsolescence are important considerations.

2.4.2 Support equipment concept
Common support equipment, instead of special support equipment, should be acquired when possible and cost effective. To reduce costs, it is important to evaluate whether or not existing support equipment can be used or can be adapted for use.
2.4.3 **Personnel integration and staff training**

Manpower issues are crucial to the supportability of many systems. It is necessary to plan for the timely training of all support personnel. Acceptable risk levels, training level needs, and manpower ratios must be addressed as supportability concerns. The training needs consist of two distinctive components: initial and on-going. Both are important in order to ensure adequate operator skill levels. Given the high level of turnover in personnel, maintaining a current operator’s skill is often a crucial issue. Repair and maintenance personnel can also turn over rapidly. Support planning must deal with these issues.

2.4.4 **Facilities**

Early planning considerations must be given to facilities because of the long lead times associated with site acquisition and allocation. Facilities that address maintenance actions must be planned with care to ensure process needs and activity work flow are supported.

2.4.5 **IT and communication resources**

This is another area where logisticians need proper preparation. What constraints are necessary in order to provide interfaces with other services? What is the trade-off when X architecture provides a desirable improvement in operational availability but denies access to Y communications network used by another service? Which IT architecture must exist or must be additionally established? It must be clear that IT resources include all aspects such as computer hardware, computer network components, network wiring, communication protocols, software packages, data security aspects and standards.

This paragraph also requires an understanding of future capabilities. Designing a system to interface with those “forecasted to exist at the time the system will be fielded” requires the engineer and logistician to be aware of the status of other related programs. How can this product interface with planned future communications architecture? The logistician must assess the impact of IT-system changes to be expected and determine necessary adjustments to the logistics structure.

2.4.6 **New organizational structures**

New organizational structure considerations have two aspects:

- Any changes to the established organizational structure at a location of the customer that must be made to support and operate the system.
- Changes in the organizational structure (eg reduction in personnel) that can be made because the system replaces old existing systems or because the new product is easier to maintain.

Organizational structure changes have impacts on available logistics support infrastructure that must be accounted for in the development of a new product.

2.4.7 **Schedule considerations**

The logistician is obviously concerned with scheduling decisions. Logistic support is a vital and integral part of any product that is fielded. Only when logistics is an afterthought would it cause delays and inefficient use of resources. If logistic considerations are interwoven with each program in each of its phases, then the supportability schedule will have been efficiently synchronized and integrated with other system schedules.

2.4.8 **Additional aspects**

Special aspects concerning areas such as packaging, storage, handling or transportation considerations can be addressed here. Unique data requirements are defined here. Logisticians must know how and when they will use the data they request, and they must be able to distinguish between “nice to have to cover possible contingencies” data and essential data. Packaging, handling, transportation, disposal and environmental impact considerations are far from the forefront for system designers, developers, and users, but they are important and potentially expensive considerations. Logisticians must understand the potential impact of these...
issues on the product from its inception, and must raise these issues whenever they have an impact on program planning.

2.5 **Detailed checklist for the creation of a customer requirements document**

For a detailed checklist how to create a CRD, please refer to Para 11.2.

2.6 **Time schedule for document creation**

To ensure that all required information is available for each of the 3 basic documents which identify product usage data, a sequence or work flow for the creation of these documents should be identified.

![Fig 1 Schedule for the creation of the basic documents](image)

2.7 **Site surveys**

Site surveys of operational units and maintenance/repair workshops can provide a significant input into the operational and customer requirements in terms of identifying existing capabilities, resources, and potential problems. Site surveys can be useful once the operational environment for the new equipment is identified in sufficient detail to determine existing operational units and repair facilities that would most likely be involved in the operations and support of the new equipment. The results of site surveys should be documented carefully and should become part of the ORD eg as an additional appendix.

2.8 **Qualification requirements**

When the customer requires a qualification process, it must be clear between customer and the contractor, at an early stage, which aspects of maintenance activities must be taken into account to achieve a qualification of the product by customer authorities. The requirements for the qualification must be defined by the customer and should be transmitted to the contractor together with the ORD.

2.9 **Certification requirements**

When a required certification process is necessary for the operation of the product in the predicted environment, special attention should be given to the collection and documentation of the maintenance activities that are a part of the certification. These activities are necessary, regardless of the costs. All aspects of maintenance activities that influence certification must have first priority. The project, in its early stages, must clarify which efforts must be considered in order to fulfill the requirements of the certifying authority. The requirements for the certification are defined by the certifying authority and must be well known to both the customer and the contractor.

The customer and the contractor must ensure that the differences between the qualification process and the certification process are evident and well known to each responsible person.
3 Establishment of product design and performance data

This chapter deals with the identification and documentation of product design and performance data/information relevant for LSA purposes.

3.1 Selection criteria concerning LSA relevant data and information

In this context, the data and information that should be considered as relevant are those that are subject to verification and control within the intended LSA process. Related criteria must be established in detail depending on the individual product that is subjected to LSA, the related contract and specifications, as well as the established ILS philosophy. Relevant information may be documented within the LSA database as requirements in order to set a goal to be verified within the established LSA process. As a general approach, the selection criteria described in the following paragraphs should be considered.

3.1.1 Selection of LSA data and information derived from contractual documents

Contractual documents should be considered carefully for LSA relevant product design and performance features that could be verified by information documented within the LSA database. Usually, overall product requirements or so-called Key Performance Indicators of importance are covered within these documents to establish mandatory goals and/or thresholds that are potentially related to contract incentives.

Examples of Key Performance Indicators:

- Specified Maximum Maintenance Man Hours per Operating Hour. This value may serve as a benchmark concerning successful maintenance design.
- Specified Maximum Mean Time to Repair paired with Specified Maximum Mean Time to Repair Percentile. These values may serve as an indication for successful design as it is related to repair within established time constraints.
- Specified Maximum Failures per Operating Hour. This value is an indication of the desired reliability of the product and inversely the maintenance workload.
- Specified figures concerning minimum availabilities. These values may serve as an indication for successful design with respect to readiness for operation.
- Testability characteristics. These values indicate the ability of the design for monitoring vital functions, detection and localization of potential malfunctions by internal means like Built in Test Equipment (BITE) and overall test architecture.
- Minimum operational lifetime. This value indicates a minimum lifetime requirement. This should indicate any risk of falling below the established threshold.

3.1.2 Selection of LSA data and information derived from product usage data

In this context, refer to Para 2. Relevant information may be documented within the LSA database as a baseline reference.

Examples:

- Annual Operating Requirements (AOR) with its measurement base
- Number of operating locations to be considered
- Number of systems to be operated at each location
- Maintenance levels to be established at each location
- Maintenance personnel available at each location (number of persons by trade and skill)
3.1.3 Selection of LSA data and information derived from design and performance specifications

In this context, refer to Chap 5. LSA relevant parameters influencing the design and performance of the product may be documented within the LSA database for verification and/or control purposes.

Examples:
- Specified Mean Time Between Failures (MTBF) together with the related measurement base indicating the minimum value
- Reliability growth
- Specified testability features concerning BITE
- Specified maximum allowable time for replacement
- Specified Maximum Mean Time to Repair paired with Specified Maximum Mean Time to Repair Percentile. These values may serve as an indication for successful design related to repair within established time constraints.

3.1.4 Selection of LSA data and information relevant for product certification and verification contained in other product documents

LSA relevant information may also be derived from other documents originated by the design department, Configuration Management (CM), safety, stress department or from the customer's documents.

Examples:
- Special operation and/or repair limitations (e.g., temperature ranges, anti-static protection requirements, clean air repair conditions)
- Delivery plans
- Validity information (e.g., version applicability)
- Hazardous classification of failures
- Storage limitations and/or requirements
- Criticality classification of specific parts
- Scheduled requirements (e.g., established time limits, overhaul requirements)

3.2 Influence of overall LSA strategies and principles on LSA data selection

Prior to the determination of relevant product design and performance data, an overall LSA strategy should be established as part of the tailoring of the LSA program. The tailoring processes are dependent on the project’s complexity, value to business units (contractor), and known risk factors. These constraints and performance requirements are then balanced between the required analysis efforts, time, schedule, and allocated budget with a cost/benefit that is best for the program. These requirements, issues, and constraints are then reviewed and evaluated at the Guidance Conference (GC) to achieve a consensus between all participants. These findings, analyses and resulting agreements are documented in the Guidance Conference Document (GCD).

This tailoring activity is an iterative process and should be repeated in each program phase. Results and lessons learned from each prior phase should be a part of and included in the tailoring analysis for each of the following phases.

3.2.1 Procedures and principles influencing selection of LSA data

The overall LSA strategies and principles may influence the selection of product design and performance data in relation to the LSA process. Examples for supportability needs to be considered in this area are as follows:

- Pre-determined two level maintenance concept

  Maintenance is mainly limited to only two maintenance levels, one for the replacement of an item and the other for its repair. This will help avoid requiring duplicate inventory at
many repair facilities. The two level maintenance concept is normally preferred because of overall cost considerations if the equipments (LRU) to be removed and replaced at the operational site have a low failure rate and a high built-in-test detectability rate, and a responsive supply chain between the operational sites and suppliers is configured.

With this concept, LSA data are limited to pre-determined Maintenance Levels (ML).

- Repair concentrated on one certain ML’s
  Repair is mainly limited to user sites in order to achieve maximum autonomy, independent of cost-effectiveness.

With this concept, repair option data are limited to the pre-determined ML.

- Limited Repair at ML 1 and/or 2
  The corrective maintenance action may be limited to an item exchange (no in situ repair) in order to shorten any operational downtime.

With this concept, maintenance task are limited to pre-determined criteria.

- Single source principle
  When major repair is necessary, the supplier of the item should be most-favored since eg related experience, personnel and equipment are available and ready.

With this concept, repair data are limited to the supplier information.

- Interim support concept
  In order to acquire experience and reduce risks prior to final Maintenance Concept (MC) decisions, an interim support phase may be established.

With this concept, MC data are limited to preliminary information.

- Commercial off-the-shelf (COTS) concept
  When equipment already available on the market is used, related conditions must be accepted.

With this concept, data have to reflect related supplier information/conditions.

### 3.3 Acceptance rules concerning the verification of values

Clearly defined acceptance rules should be established in order to avoid uncertainty during the verification process. These rules will allow accurate acceptance or rejection of the results for LSA relevant product design and performance data.

#### 3.3.1 Category of measurement values

The type of requirement should be stated regarding the importance of the related values such as:

- Mandatory values
- Objectives
- Thresholds (minimum or maximum values)

#### 3.3.2 Identifying tolerances

For each identified value, the associated tolerances should be expressed.

- Tolerances for the dedicated value expressed by the requirement for one single value
− Tolerances concerning a group of similar values, for example, possible compensating of failure rates within a given area of items under analysis in order to meet the failure rate of the group instead of the single items, eg by following additional constraints

3.3.3 Establishment of acceptance criteria
Measurable acceptance rules should be established for each specified value along with its related requirement. These rules should identify the basic conditions (eg fulfillment of minimum required values based on sufficient statistic confidence levels) that lead to the acceptance or rejection of the values documented within the LSA database.

In addition, the consequences of the established status information should be stated clearly:
− Status of the Item under Analysis (IUA) indicating the acceptance of the documented value
− Status of the IUA indicating the rejection of the documented value (with related justification).
− Status of the IUA indicating the conditional acceptance of the documented figure (eg "in general acceptable but requiring minor rework")

3.3.4 Establishment of special rules
When special rules are established, the related conditions and related values have to be well defined in order to avoid uncertainty:
− Establishment of penalty regulations concerning failed specified LSA significant data
− Establishment of rewards in case of exceeding specified LSA significant data

3.4 Criteria and procedural aspects for verification of projected values
Finally, the criteria for verification of data and/or other information being subject to verification should be established as follows:

3.4.1 Determination of measurable target values
For data elements under consideration the related projected value range should be established and documented within the LSA database as a requirement by considering the original target along with its established tolerances and relevant acceptance rules, if any. The projected value range may be allocated to a single item and/or a group of dependent items covered by relevant acceptance rules in order to establish the related acceptable values that fulfill the requirements.

3.4.2 Verification of the individual projected values
The relevant data and other information documented in the LSA database derived from the analysis process should be compared to the associated acceptable values. The result should be reported to the analyst and/or involved management indicating whether or not the actual LSA values consistent with the projected values.

3.4.3 Verification method
Projected values and their method of verification should be agreed upon:
− Verification by furnishing analytical proof (documented within the LSA database)
− Verification using an analytical method based on appropriate tests (eg at a test rig)
− Verification by demonstration (on a prototype or serial versions of the product)
− Verification under the rules of certification and/or demonstration programs
− Verification by trial sessions (eg performed by customer's staff)
− Verification during long term exercises (eg during a defined "maturity phase")

3.4.4 Verification for special purposes
When verification for special purposes are necessary (eg in order to gain certifications, qualifications, licenses), particular rules may be established and required.
3.4.5 **Update of status codes**
Depending on the verification results, the related status information should be linked to the verification result by following the rules established for status allocation.

3.5 **Customer involvement**
The selection of product design and performance data subject to verification and control within the intended LSA process should be agreed upon by the customer during the LSA GC and documented in the GCD. This also applies to the agreement of related rules and projected values.

3.5.1 **Checklist for the LSA guidance conference**
The checklist should contain all relevant proposals for design and supportability requirements, eg:

- Have LSA relevant product design and performance data been identified?
  - List of selected LSA data and information derived from contractual documents
  - List of selected LSA data and information derived from product usage data
  - List of selected LSA data and information derived from design and performance specifications
  - List of selected LSA data and information contained in other product documents
  - Have selected LSA data and information subject to certification and verification or other special requirement (eg for license purposes) been noted accordingly?

- Have acceptance rules concerning the verification of relevant values been identified?
  - Have measurable target values been identified for the selected data/information?
  - Have the category of measurement figures been stated against the selected data/information (eg mandatory values, goal values, thresholds)
  - Have tolerances been defined for the selected data/information?
  - If applicable, have compensating rules been established?
  - Are projected values acceptable/agreed to by the customer?
  - Are the consequences of acceptance/agreement concerning status information identified?
  - If applicable, have penalties and/or rewards regulations been established?

- Are overall LSA strategies and principles established that influence these?
  - Identify the preferred maintenance concept (eg is two level maintenance mandatory?)
  - Identify the supply chain management support concept
  - Have the repairs focused on one maintenance level (which level)?
  - Is repair at Maintenance Level 1 and/or 2 limited?
  - Is single source principle determined?
  - Is an interim support concept applicable?
  - Is COTS concept to be considered?

3.5.2 **Documentation of the results of the LSA GC**
The details of the LSA CG checklist must be documented as part of the GCD.

4 **LSA guidance conference**
The LSA GC should be the central event with participation of management staff and specialists on both sides, customer and contractor. At this conference, the binding agreements for the performance of the LSA process must be established. To ensure best performance of this conference, it is necessary to have prepared inputs and to have a clear expectation of the results and final agreements. It is strongly recommended to have checklists for the LSA GC preparations and expectations.
Note

Never accept "to be determined" (TBD) as an input for any subject to be discussed in the LSA GC. Discussions should always be based on concrete proposals prepared by specialists, on how to realize the required task. Evaluate the advantages and disadvantages (cost/benefit) of the alternatives under consideration. Provide documentation of all alternative analysis and decisions.

4.1 The transition from a request for proposal to the LSA guidance conference

It should be pointed out clearly that the majority of essential decisions influencing the LSA effort have to be negotiated before the LSA GC. Usually, the LSA process begins while creating an offer and will be similar to the final version contained in the corresponding contract. This applies to any LSA significant aspect within the contract (eg related Statement of Work) as well as for contractual details such as eg deliverable items, indispensable specified values or major milestones. This implies a series of investigations to be done prior to the contractual offer (eg identification of LSA activities considered as mandatory, recommended or voluntary, depending on early strategy judgment and/or the kind of systems and equipments to be assessed).

Fig 2 LSA tasks in conjunction with the preparation of an offer/contract

The LSA GC serves as a vehicle to communicate to the customer the work that will be done in detail, along with the associated rules and time schedules, based on the contractual requirements and further agreements as noted above. The LSA GC also clarifies any questions the customer might have regarding the effort. Nevertheless, changes to the LSA work effort, to a certain extent, must be allowed during the LSA GC without the need of contract changes and changes in the cost of the effort. Considering the iterative nature of LSA activities, customization (tuning) must be possible in order to be flexible.
4.2 LSA business process at a glance

<table>
<thead>
<tr>
<th>CUSTOMER ACTIVITIES</th>
<th>CONTRACTOR ACTIVITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basis / Starting Point:</td>
</tr>
<tr>
<td>Contract or MoU / Top Level Requirement Specification</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Establishment of</th>
<th>Preparation of LSA</th>
<th>Establishment of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Usage Data</td>
<td>Guidance Conference</td>
<td>Product Design &amp;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Performance Data</td>
</tr>
<tr>
<td>General usage aspects</td>
<td>Candidate Item List</td>
<td>General Product</td>
</tr>
<tr>
<td>Operational Requirements</td>
<td>Data Element List</td>
<td>Design Aspects</td>
</tr>
<tr>
<td>Customer Requirements</td>
<td>Programme Plan</td>
<td>General Performance</td>
</tr>
<tr>
<td></td>
<td>- DRAFT -</td>
<td>Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DRAFT -</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GC Document (GCD) - DRAFT -</td>
</tr>
</tbody>
</table>

**LSA Guidance Conference**

<table>
<thead>
<tr>
<th>Candidate Item List</th>
<th>Data Element List</th>
<th>Programme Plan</th>
<th>GC Document (GCD)</th>
</tr>
</thead>
</table>

**Assessment of Candidate Item selection and identification**

Agreement to CIL

Agreed?

yes

no

**Clarification of non-agreement and/or modification of Candidate Item List**

Agreement to recommended analysis activities

Agreed?

yes

no

**Clarification concerning non-agreement and/or identification of solution how to proceed**

**Recommendation of analysis tasks per Candidate Item**

**Delivery of recommended analysis activities per CI**

**Delivery of CIL to the customer**

**Agreement to CIL according to agreed rules**

**Establishment of breakdown according to agreed rules**

---

Fig 3 Flowchart of LSA business process at a glance (Sheet 1 of 2)
4.3 Documents and Information for LSA guidance conference

In the following paragraph you will find a list of applicable documents relevant for the LSA GC. This list should give you a guideline for which information should be documented in which way. The particular project may add additional aspects or skip some information. The following figure should give a summarized overview of the LSA GC concerning the required documents:

**Note**

It should be kept in mind that the majority of essential decisions influencing the LSA effort should be negotiated before the LSA GC and documented in the contract. This applies to any LSA relevant aspect within the statement of work and the corresponding parts of the commercial offer, as well as for contractual details such as deliverable items, indispensable specified values and major milestones. This mandates a series of investigations to be done prior to the contractual offer (e.g., identification of LSA-analysis activities considered as mandatory/recommended/voluntary - depending on early strategy judgment and/or the kind of systems/equipments to be assessed).
4.3.1 Checklist of documents for the LSA guidance conference

In the following table an overview of required information is given to prepare for a LSA GC.

**Table 2 Required input documents for LSA GC**

<table>
<thead>
<tr>
<th>Type of document</th>
<th>Content</th>
<th>Responsible for creation</th>
</tr>
</thead>
<tbody>
<tr>
<td>General usage aspects</td>
<td>Strategic aspects of the product design and performance data, which are described in Para 3. Customer</td>
<td></td>
</tr>
<tr>
<td>Contractual documents</td>
<td>Contractual decisions, which can influence the LSA process, must be available at the LSA GC and must be considered. Customer and contractor</td>
<td></td>
</tr>
<tr>
<td>Operational Requirements Document (ORD)</td>
<td>Details about the usage of the systems in the planned scenario at all operational locations. These details are described in Para 2. Customer</td>
<td></td>
</tr>
<tr>
<td>Customer Requirements Document (CRD)</td>
<td>Details of customer requirements regarding the operational scenario described in the ORD. These details are described in Para 2. Customer</td>
<td></td>
</tr>
<tr>
<td>Candidate Item List (CIL)-DRAFT</td>
<td>A list of Breakdown Elements (BE) to be considered as potential candidates for a LSA and for the performance of any logistic analysis activities Contractor</td>
<td></td>
</tr>
</tbody>
</table>
4.3.2 Checklist of documents and information from the LSA guidance conference

The result of the LSA GC must be an integrated consensus of measurable rules for guiding the LSA process. These rules, checklists and documents provide a clear road map of needs and expectations between the contractor and the customer. This collection of official documents must be agreed to and signed by the customer and the contractor. These documents should contain the following (as a minimum):

Table 3 List of output documents of the LSA GC

<table>
<thead>
<tr>
<th>Type of document</th>
<th>Content</th>
<th>Responsible for creation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminary Candidate Item List (CIL)</td>
<td>List of LSA candidates containing all BEs that have been selected for logistic analysis activities, including the analysis activities to be performed for each BE.</td>
<td>Customer and contractor</td>
</tr>
<tr>
<td>LSA Program Plan (PP)</td>
<td>Initial release of the preliminary LSA Program Plan (PP). LSA GC consensus of attending members.</td>
<td>Customer and contractor</td>
</tr>
<tr>
<td>LSA GC Document (GCD)</td>
<td>Initial release of the LSA GCD. LSA GC consensus of attending members.</td>
<td>Customer and contractor</td>
</tr>
</tbody>
</table>

4.4 LSA program plan

The LSA PP is a crucial output document of the LSA GC. It should be organized with general sections and descriptive subsections. These are followed by the appendices. In the general section, the following aspects should be covered:

- Management structure of the program for both the contractor and customer
- Time schedule of the program and meeting calendar
- Definition of milestones
- Responsibilities and reporting levels
- Change process (categories and levels of authority)
- Risk management
- Work share agreements

The following figure shows an example of the structure of a LSA PP:
### CONTENT

<table>
<thead>
<tr>
<th>Project XXX</th>
<th>LSA Program Plan</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>General</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Scope</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>General program definitions</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Management structure of the program on customer side</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Management structure of the program on contractor side</td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>Time schedule of the program</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>Definition of milestones</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>Meeting calendar</td>
<td></td>
</tr>
<tr>
<td>3.6</td>
<td>Responsibilities and reporting levels</td>
<td></td>
</tr>
</tbody>
</table>

### APPENDICIES

- **Appendix A** Purpose of LSA and of logistic analysis tasks
- **Appendix B** System breakdown methodology
- **Appendix C** Breakdown depth
- **Appendix D** Rules for LSA candidate selection
- **Appendix E** Candidate Item List
- **Appendix F** Rules for analysis tasks selection
- **Appendix G** Rules for performance measuring and verification
- **Appendix H** IT aspects

---

**Fig 6 Example of content of a LSA Program Plan**

The appendices will provide additional clarity and specific needs which supports the content in the subsections.

#### 4.4.1 Purpose of LSA and logistic analysis activities

As a basic result of the LSA GC, the contractor and customer must agree to the principles of how to use the data coming from the logistic analysis activities. The documentation of data within a logistic database must define the purpose of collecting the data. Therefore, it is strongly recommended to carefully select which data elements will be documented in a logistic database and then link the data with its corresponding purpose. This also applies to the logistic analysis activities. The selection must consider technical and economical aspects, especially for very extensive analysis. Examples can be the following:

- Detailed or simplified Level of Repair Analysis (LORA)
- Optimization methods, such as via simulation runs
- Detailed Scheduled Maintenance Analysis (SMA)

#### 4.4.2 Hierarchical product breakdown methodology

In this appendix to the LSA PP, all the results of the procedure described in Para 5 for the establishment of a product breakdown methodology must be documented. It must be clarified between the customer and the contractor, how the Breakdown Elements (BE) are to be identified, eg using system hardware identification according to S1000D, Logistic Control Number (LCN) in accordance with MIL-STD 1388-2B, or any other identification system preferred by the customer.

If it is necessary to use different breakdown methodology (physical and functional) within a program in parallel, it must be clarified whether and how to interconnect these different breakdowns. Additionally, the purpose of its use must be documented (eg a functional breakdown).
Note

It is recommended that detailed examples in the appendix should be included, covering all possible cases (eg it must be clarified how software will be incorporated within the hierarchical breakdown, it must be clarified how to cover different end item variants).

4.4.3 Breakdown depth for the item under analysis

In addition to the decisions required for the types of breakdown, the breakdown depth is a crucial attribute of the breakdown itself. The following aspects must be considered for a final decision:

- Is a full breakdown of the item required to identify all relevant spare parts and is this breakdown type effective and applicable?
- Is a breakdown that only contains LRUs applicable and effective?
- What depth of breakdown is required to identify all spare parts for a repair concept at the authorized level?
- For what purpose shall a functional breakdown be established (eg for SMA) and how is the functional breakdown linked to the physical breakdown?

4.4.4 LSA candidate selection criteria

The criteria for LSA candidate selection as described in Para 6 must be documented in an appendix of the LSA PP. A decision flowchart should also be part of the appendix.

4.4.5 Candidate item list

As an input to the LSA GC, a draft of the CIL must be prepared as an input document by the contractor. During the GC, this draft must be discussed between the contractor and the customer, resulting in an agreed upon CIL. The CIL will be an official output document of the LSA GC.

4.4.6 Potential analysis activities for candidate items

The CIL does not only contain the selection of the LSA candidates from the product breakdown, it should also contain the analysis activities to be performed for each LSA candidate. In addition to the analysis activities selection, the depth of the analysis should be clarified. For example, in the area of the Maintenance Task Analysis (MTA), a decision must be made about how deep the description of the maintenance task should be documented or whether it is necessary for the project to identify personnel requirements. Finally, the CIL should serve as a LSA program overview, in which the complete effort is summarized. Additionally, the CIL can be used as a helpful management tool, documenting the progress of the LSA program.

Note

As a final output from the LSA GC, a CIL must be generated. It is recommended that this list is designated as a reference table and part of the projects master data which is shared and synchronized across the project’s master data management system. It can be used as a monitoring tool to supervise the progress of the project. This will avoid duplicating data, such as breakdown information, in external lists which are not synchronized to cleansed master data.

4.4.7 Performance measurement and verification of LSA activities

The performance of LSA activities should be continuously evaluated and documented eg by status information within the logistic database. The criteria, when an analysis activity is fulfilled or when a LSA candidate is completely documented within the logistic database, must be clear and documented in a list of acceptance criteria that are a part of an appendix within the GCD. The process and the rules of performance measurement and verification must be clear. For each delivery of any analysis result, these rules must be obeyed by both the customer and the contractor.
4.4.8 IT aspects

As a general recommendation, using different software packages for the same analysis activities at different locations should be avoided. Integration and harmonization processes can be extensive and are a potential risk factor. However, many projects currently use different software packages because of existing licenses, contractual constraints or IT environment necessities. In this case, for any task to be performed, suitable software packages compatible between all partners should be found.

The decisions about software packages must be documented at the LSA GC. Changes of software releases during a program must be harmonized and agreed to between all impacted partners.

4.5 Guidance conference document

The GCD is another crucial output document of the LSA GC. It contains the implementation rules of the LSA process on reporting, data element definitions, data exchange procedures and selection of software packages to perform analysis activities.

In the following figure, an example for a possible structure of this document is given.

<table>
<thead>
<tr>
<th>Project XXX</th>
<th>LSA Guidance Conference Document</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CONTENT</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>General</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Scope</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Implementation rules</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Simple data element list (DEL)</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Reporting process</td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>Definition of required reports</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>Data exchange processes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>APPENDICIES</td>
<td></td>
</tr>
<tr>
<td>Appendix A</td>
<td>Detailed data element list (DEL &amp; input instructions)</td>
<td></td>
</tr>
<tr>
<td>Appendix B</td>
<td>List of selected software packages per analysis task</td>
<td></td>
</tr>
<tr>
<td>Appendix C</td>
<td>Report examples</td>
<td></td>
</tr>
<tr>
<td>Appendix D</td>
<td>Mathematical appendix</td>
<td></td>
</tr>
</tbody>
</table>

Fig 7 Example of content of a Guidance conference document

The GCD should be organized with a general section and a number of appendices, which are described in the following paragraphs. In the general section, the definition of the implementation rules should be covered:

- Data Element List (DEL):
  List of all the required data that should be recorded in a logistic database.

- Reporting process and definition of required reports:
  All report content and format must be defined clearly. Any data elements required for the reports must be available and accessible. All contracting parties will agree on content, format and derived calculations. Any business rules necessary to calculate and/or select source data will be identified and documented.

- Data exchange process:
For a proper exchange of information, a data exchange process must be established and harmonized between the customer and the contractor. Especially in programs with several partners on both sides, customer and contractor, a clear process for data integration and harmonization is required. A time schedule should be agreed to during the LSA GC and it must be clarified the consequences of missing the time schedule.

**Note**

A detailed description of the recommended data exchange format by the usage of ASD DEXs is given in [Chap 20](#).

### 4.5.1 Data element list and input instructions

The detailed DEL describes the context and description of the data being captured in a LSA database. This includes syntax (if required) and allowed codification. Also, any calculation methods of numeric values should be described in a mathematical appendix.

To support the initial selection of data elements, it is recommended to create a simple draft DEL as an input to the LSA GC. The justification for the project's LSA DEL selection is based on the following questions:

- Is the data element required for the proof of any specification values?
- Is the data element required for the calculation of any logistic parameters?
- Is the data element required for the following disciplines such as technical documentation, materiel support, identification of special support equipment, and identification of facilities or training requirements?
- Is the data element required for the performance proof of the LSA itself?
- Is the data element required to document the results of the logistic analysis activities?
- Is the data element required for any report, necessary for the customer and/or the contractor?
- Is the data element a special interest for the customer/contractor (e.g., internal usage of contractor)?

**Note**

It is strongly recommended to select the data elements based on a logical, traceable project requirements need. Listing of any data elements that do not link to a requirement or the development (calculation) of a requirement should be a candidate for elimination.

The DEL must be agreed to during the LSA GC and will be a part of the official GCD. Nevertheless, it is to be considered as a living document. The DEL should be updated, when additional needs arise during program maturation (e.g., matured project design as indicated by contractor and customer).

### 4.5.2 List of selected software packages

In the GC, the decision should be made (or at least prepared), on which software packages will be used by the contractor and the customer to support the required analysis activities. The selection of software packages should be contractual. An upgrade or a complete change of software packages within the project must be investigated carefully, including all participants involved with such changes in the IT environment.

### 4.5.3 Reports examples

To guarantee a common understanding of reporting, examples should be part of the GCD. These examples should convey to the conference members: formatting, sorting, filling and calculations.

### 4.5.4 Mathematical appendix

To complete the implementation rules, each method of calculation should be documented. The calculation of values should be evaluated by both, contractor and customer, and should be
available for review by those who have a need. For this reason, the mathematical appendix should cover two basic aspects:

- Mathematical equations and their explanations (with list of mathematical abbreviations and symbols)
- Sources and business rule logic from which the data is obtained for calculation

4.6 Candidate item list
The creation of the CIL including the associated business rules is described in detail in Para 5 and Para 6.

4.7 Contractual documents
Contractual conditions are the basis for a project. All contractual decisions that were made prior to the LSA GC must be carefully considered. If these decisions impact the ILS process itself, the selection of the logistic analysis activities will be crucially influenced.

Another contractual consideration is the significance of the output documents of the LSA GC. The supporting LSA GC documentation should support contractual requirements and needs for both the contractor and the customer.

5 Establishment of breakdown according to agreed rules
A systematic breakdown of a product is essential to the LSA process and provides the following:

- A clear understanding of how the product is structured with respect to systems, subsystems, functions, hardware and software components. This also applies to peripheral equipment that may be included in the LSA process, for example complex training equipment and support equipment.
- A clear relationship between the included hardware elements and their possible realizations in terms of manufactured hardware parts.
- A clear relationship between the included software elements and their possible realizations in terms of actual software releases.
- Enable the allocation of unique or unified identifiers. This enables the establishment of interfaces between design departments including configuration sources, LSA and the logistics disciplines such as technical documentation, materiel support, training and support equipment.
- Enable an appropriate level of detail for the selection of LSA candidates.
- Identification of product variants and product configurations.
- Enable CM with respect to how a specific revision of the product breakdown relates to the correct revisions of its corresponding ILS end products. It also enables the CM of source data on which the product breakdown and its ILS products is based, including documents, design information and product usage data.

Different breakdowns of the product can be defined for different purposes. In many projects it is common to have a design breakdown of the product being defined in some kind of Product Data Management (PDM) system. The same product often also has a LSA breakdown, as well as a separate breakdown for the recording of operational and maintenance data feedback. Different breakdowns for different purposes often results in mismatches in between the different breakdowns, and complicates to communicate product related data in between the different disciplines.

This chapter describes both, the situation where there is need to define a separate breakdown of the product from a LSA point of view, as well as the situation where the LSA process can reuse the breakdown as defined by design.
5.1 Definitions of general terms

5.1.1 Product/end item
The product (or end item) is a final combination of systems, subsystems, component parts/materials and software (e.g., ship, vehicle, machines, aircraft or other complex technical systems). The product represents the top level (root) of any hierarchical product breakdown.

5.1.2 Product variant identifier
The product variant identifier is a code, which uniquely identifies the product variant. It is recommended to use the product variant identifier in conjunction with other identifiers within the entire ILS process.

5.1.3 Product breakdown
A product breakdown consists of the product itself, and the hierarchical listing of, for example assemblies, subassemblies, and components that can be disassembled, reassembled or be replaced. Depending on the breakdown philosophy, a product breakdown can also consist of systems, subsystems, functions and zones. Examples of types of product breakdown are described in Para 5.2.

It should be pointed out that the required depth of a product breakdown in a LSA database normally will be different from, for example, the one being required for materiel support or production. Nevertheless, breakdown rules concerning breakdown syntax should be consistent within a project.

5.1.4 Breakdown element and breakdown element identifier
The BEI identifies the individual BEs that is defined within a product breakdown. It is strongly recommended that the establishment of the BEs and their BEIs is co-ordinated with the other logistic disciplines and with the design department. The BEI should help to uniquely identify any item that is a part of any logistic analysis activity. Para 5.2, describes the different types of systematic product breakdowns. Depending on the type, different rules for the establishment of the BEI should be addressed.

5.1.5 Breakdown element revision
A BER defines a new release of a BE. BER captures the development progress (often referred to as design iterations). The use of BER supports CM and especially the synchronization between changes that are being introduced from product design, and its consequences in the LSA results.

Note
The BER must not be mixed up with methods to document alternative components within a product breakdown as, for example, by Alternate Logistic Control Number (ALC) in MIL-STD 1388-2B. The situation, that an equipment which has to be installed at a certain position within the product breakdown has two different realizations (e.g., two different equipments of different manufacturers which have different maintenance concepts) is covered in S3000L by the means of related realizations in terms of parts (hardware or software). These parts also can be LSA candidates against which the results of the technical/logistic analysis activities can be documented.

5.1.6 Replaceability aspects
Some common terminology has been established in the area of logistics concerning the replacement of equipment/components. These are given in the following table:

<table>
<thead>
<tr>
<th>Term</th>
<th>Full name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRU</td>
<td>Line Replaceable Unit</td>
<td>Directly replaceable on product level</td>
</tr>
</tbody>
</table>
SRU  Shop Replaceable Unit  Replace on LRU or SRU level, but not directly replaceable on product level under normal circumstances. This means, the superior component (LRU or SRU) may need to be removed before the concerned SRU can be replaced.

ASSY  Assembly  Artificial level to structure the breakdown

**Note**
In Para 5.6.4, the table above will be detailed. In addition to item replacement, repairability aspects are considered.

### 5.1.7 Parts
Parts are the actual realization of BEs defined in a product breakdown, and can be either hardware items or software packages.

### 5.1.8 Product related applicability – Management of product variants and configurations
Product related applicability (often also referred to as effectivity) is used to reflect the configuration information of a BE with respect to its allowed use in different product variants. It covers the need to limit the applicability of any item in a product breakdown to special product variants. An example for the usage of BE and applicability is given in Para 5.6.2.

**Note**
It is strongly recommended that the CM aspects between the customer and the contractor, and the meaning of CM must be clarified by both the customer and the contractor. In general applicability should be used for management of product variants. In this context, different product variants should be designated via applicability to the BEs.

It is recommended that the LSA database is not used as a CM tool for the as-built situation (eg to enter the configuration of each end item delivered to the customer, into the LSA database).

### 5.2 Types of product breakdown
A breakdown of a product can contain both functional and physical aspects. These two approaches are illustrated by the examples in Fig 8 and Fig 9. Any LSA database should be able to document both methods and combinations thereof (so called hybrid breakdowns).

The interrelationship of an item with other items at the same indenture level and the next higher indenture, as well as its breakdown items (eg attaching parts), can be reflected by the BEI and BE realizations or via explicit parent child relationships, respectively. To reflect the structure of a product the following methods are recommended, depending on the requirements of the project.

− A pure functional breakdown
− A pure physical breakdown
− A mixed breakdown, functional and physical in one breakdown tree (often referred to as hybrid breakdown)
− A pure functional and pure physical breakdown parallel, with cross-mapping

Special attention should be given to software. In modern state-of-the-art systems, software use is increasing. This should be considered in any logistic analysis. It should be possible to integrate loadable software packages into the breakdown, if the installation/de-installation of software is maintenance relevant. The assignment of software to hardware should be clear within the breakdown. Configuration changes due to software modification are another aspect of the breakdown. If hardware will not be modified, but software integrated in the hardware will change, this situation has to be documented in a product breakdown.
5.2.1 Functional breakdown
A simple functional breakdown should start on top level of the product as the root of the breakdown tree. The different functions of the product should be documented from the main functions down to the sub-functions to the required depth. Within the pure functional breakdown, no hardware elements should be found.

![Functional breakdown diagram]

*Fig 8 Simple functional breakdown*

5.2.2 Physical breakdown
A simple physical breakdown should also start on top level of the product as the root of the breakdown tree. As a general principle, in each physical breakdown it should be possible to structure the breakdown by the usage of assemblies. As shown in *Fig 9*, this is realized on the second indenture level by introducing two different tank assemblies.

![Physical breakdown diagram]

*Fig 9 Simple physical breakdown*

Below the product itself, the actual hardware elements are documented. Within a pure physical breakdown, no function should be found.

5.2.3 Zonal breakdown aspects
A zonal breakdown can be accomplished by sequencing all the zones defined for the product. It should be pointed out that the required depth and structure of a zonal breakdown established in a LSA database may be different from, for example, the one being defined for product design. Nevertheless, breakdown rules concerning breakdown syntax should be consistent within a project.

Zones as BEs are typically required for the documentation of maintenance activities which are related to the physical area represented by a zonal BE (eg scheduled zonal inspection). To describe the content of a zone, it should be possible to establish relationships between zonal BEs and physical BEs within a LSA database.
5.2.4 Installation location of equipments/components and multiple installation

In general, the installation location is the primary information within the physical breakdown. Multiple installations of equipment/components can be considered in two different ways:

- One BE with quantity information (e.g., quantity per system)

![Fuel system diagram](ICN-B6865-S3000L0010-001-01)

*Fig 10 Breakdown methodology - Simple parent-child philosophy*

In the example *Fig 10*, the same type of fuel tank is installed three times within the system. The deeper breakdown of the fuel tank is always the same. In some cases, it can be sufficient to have only one BE for the three tanks within the breakdown with the additional information of the quantity per system.

- Several breakdown elements

![Fuel system diagram](ICN-B6865-S3000L0011-001-01)

*Fig 11 Breakdown methodology - Extended parent-child philosophy*

In this example *Fig 11*, each fuel tank is considered as a separate BE. This method should be used for typical LSA purposes. The installation location is decisive (e.g., the installation or the removal can be totally different depending on the installation location).

5.2.5 Mixture of functional and physical breakdown methodology

An established product breakdown approach is a mixture of functional and physical aspects within one breakdown tree, as shown in *Fig 12*. Typically systems and subsystems are of functional character, equipment and components are physical breakdown elements. Nevertheless, a system need not be one concrete function but more a collection of components that have a set of common characteristics e.g., an electrical system. In this case, a system is built from a set of functions/components, but there is still a system BE which is neither a concrete function nor a physical item.
Fig 12  Breakdown methodology - Mixture of functional and physical methodology

If a mixture of functional and physical breakdown methodology is used, it can be useful to first generate a functional structure of the product dividing it into several basic systems and subsystems (eg engine system, electrics, structure, hydraulic system, fuel system). Depending on the magnitude of a system, the indenture level where the documentation of real hardware BEs begin is not always the same. The typical breakdown path, system → subsystem → equipment → component is not always fixed, so the indenture level at which LSA candidates are located can be different from system to system.

5.2.6 Parallel usage of functional and physical breakdown methodology

If it is required to establish both functional and physical breakdowns in parallel for different analysis purposes, each breakdown tree should be established as separate breakdown structures. The decisive factor is the generation of an effective connection between these two breakdown trees. Physical BEs must be linked to functional BEs.

This methodology can be a good basis for troubleshooting analysis. Based on a functional breakdown, troubleshooting should be able to identify the reasons that certain functions are not available. For this reason, the analyst must know which hardware elements could be responsible for the missing function. The parallel usage of functional and physical breakdowns also allows assigning hardware to functions. For example, the function "fuel distribution" from Fig 12 does not only contain typical elements such as fuel pipes, but also electric power supply elements. However, electric power supply is documented in another area, but it must be assigned to the function “fuel distribution”.

The cross functional interconnection between different breakdown methodologies should be considered in a logistic database structure also.

5.2.7 Family concepts in physical breakdown methodology

To reduce efforts in the establishment of a physical breakdown, family concepts should be used whenever possible. The following aspects can be considered when deciding whether or not to use a family concept:

- Similar items with common maintenance concepts can be grouped by family BEI concepts (eg harnesses of the same type, pipes of the same type, and structural items of the same type).
- Standard spare parts can be grouped by one BEI family
However, the grouping of items under one BEI needs a possibility to document the single group elements in a certain place within the LSA database. The required logistic information should be collected within the group BEI.

5.3 Background for different approaches to establish a product breakdown

As mentioned in Para 5.2, a product breakdown can be defined for different purposes. The respective breakdown is traditionally often defined and used by a specific discipline. However, the correlation between these breakdowns is often non-existent, which means that there is a problem to exchange data and share experience in between different disciplines. Examples are:

- BE that represents a position in a product are often identified in different ways within different breakdowns.
- The breakdown philosophy (methodology) is often different from each other, which makes it hard to identify similarities and differences in between the different breakdowns.

Product breakdown approaches are driven by different requirements. The working area of design and development needs another kind of product breakdown than eg support engineering for technical/logistic analysis activities. For that reason there are specific views on how to structure a product breakdown. The intention of S3000L concerning this subject is to be as clear as possible but also be flexible when required. For that reason the integration of the following two main approaches to product breakdown into one data model which supports both alternative solutions is a main goal of S3000L.

- Approach 1:
  Traditional LSA product breakdown by means of structured BEI (eg similar to LCN/ALC structures as described in MIL-STD 1388-2B or DEF-STAN 0060). Functional and physical areas are integrated into one common product breakdown. Indenture levels are defined in an implicit way by the syntax of the specific identifier (BEI).

- Approach 2:
  Design oriented or PDM based product breakdown by means of explicit parent-child relationships. These breakdowns are driven by relations of hardware components. The identifiers of the components are not structured by syntax to document indenture levels as in approach 1
The following figure gives a graphical overview of the main differences of the two basic product breakdown approaches described above:

Classical LSA Product breakdown

- Classical LSA Product breakdown uses an implicit breakdown approach. An implicit breakdown approach defines the position of the BE by the syntax of its BEI. This methodology can be established by using existing standards like MIL-STD 1388-2B or DEF-STAN 0060 respectively. In these basic standards the breakdown elements are represented by a BEI known as the LCN. Another approach can be the usage of S1000D Standard Numbering System (SNS). Central for both approaches is that the position for the respective BE within the overall breakdown is determined by its BEI syntax.

- The consequence for LSA process is that the design breakdown needs to be quite stable before the LSA work can start. One reason for this is that once the LSA product breakdown has been established, it's been nearly impossible to introduce intermediate levels into the breakdown.

- Another consequence using BEIs as the way to organize breakdown hierarchies is that a BE cannot be reused in between different products in a simple way.

- Since these kind of breakdowns do not distinguish between the function being performed and the part that realizes the function, there is a need to introduce so called alternative breakdown elements (ALC in MIL-STD 1388-2B and Disassembly Code Variants in S1000D). An alternative BE is just another BE, where its BEI indicates that it is alternative equipment or component to an existing BE. The BEI therefore indicates that the BE can occupy the same installation position in the breakdown as an existing BE. This means that such an alternative BE will have its own unique BEI and its unique BER.

Design or PDM-based Product breakdown

- Design or PDM-based Product breakdown uses an explicit parent child relationship. An explicit breakdown approach defines the position of the BE by the syntax of its BEI. This methodology can be established by using existing standards like MIL-STD 1388-2B or DEF-STAN 0060 respectively. In these basic standards the breakdown elements are represented by a BEI known as the LCN. Another approach can be the usage of S1000D Standard Numbering System (SNS). Central for both approaches is that the position for the respective BE within the overall breakdown is determined by its BEI syntax.

- The consequence for LSA process is that the design breakdown needs to be quite stable before the LSA work can start. One reason for this is that once the LSA product breakdown has been established, it's been nearly impossible to introduce intermediate levels into the breakdown.

- Another consequence using BEIs as the way to organize breakdown hierarchies is that a BE cannot be reused in between different products in a simple way.

- Since these kind of breakdowns do not distinguish between the function being performed and the part that realizes the function, there is a need to introduce so called alternative breakdown elements (ALC in MIL-STD 1388-2B and Disassembly Code Variants in S1000D). An alternative BE is just another BE, where its BEI indicates that it is alternative equipment or component to an existing BE. The BEI therefore indicates that the BE can occupy the same installation position in the breakdown as an existing BE. This means that such an alternative BE will have its own unique BEI and its unique BER.

Fig 14 Product breakdown - Traditional LSA approach versus PDM approach

5.3.1 Traditional LSA product breakdown

A traditional LSA related product breakdown uses an implicit breakdown approach. An implicit breakdown approach defines the position of the BE by the syntax of its BEI. This methodology can be established by using existing standards like MIL-STD 1388-2B or DEF-STAN 0060 respectively. In these basic standards the breakdown elements are represented by a BEI known as the LCN. Another approach can be the usage of S1000D Standard Numbering System (SNS). Central for both approaches is that the position for the respective BE within the overall breakdown is determined by its BEI syntax.

The consequence for LSA process is that the design breakdown needs to be quite stable before the LSA work can start. One reason for this is that once the LSA product breakdown has been established, it's been nearly impossible to introduce intermediate levels into the breakdown.

Another consequence using BEIs as the way to organize breakdown hierarchies is that a BE cannot be reused in between different products in a simple way.

Since these kind of breakdowns do not distinguish between the function being performed and the part that realizes the function, there is a need to introduce so called alternative breakdown elements (ALC in MIL-STD 1388-2B and Disassembly Code Variants in S1000D). An alternative BE is just another BE, where its BEI indicates that it is alternative equipment or component to an existing BE. The BEI therefore indicates that the BE can occupy the same installation position in the breakdown as an existing BE. This means that such an alternative BE will have its own unique BEI and its unique BER.
Alternative BEs can be result of, for example, different manufactures that are selected for a special item at a special installation place, where the respective manufactured item requires different logistic considerations (in form, fit and function, they are the same).

It’s also a fact that incorporating breakdowns from subcontractors enforces the subcontractor to adopt the primary contractors BEI philosophy, and implement the BEI philosophy into its process and documentation. This increases the cost for updating and maintaining product breakdowns that already exists.

**Note**
In traditional product breakdown concepts there is no mechanism to document revisions of a BE based on design progress. For that reason it is hard to synchronize changes coming from design since there is no way to track the revision of a BE based on development status.

### 5.3.2 PDM-oriented product breakdown

Product breakdowns as defined in, for example, today’s PDM systems, defines explicit parent-child relationships between its constituent breakdown elements. This means, that a breakdown can be established without having a structured methods for the identification of the respective breakdown element. Using explicit parent-child relationships requires that each breakdown element documents:

- Relationships to its child BEs
- Each parent-child relationship can include information about the quantity of child elements and installation location

**Note:**
The unique BEI for BEs as defined by design are often generated by the PDM system as a random code.

BEs are often defined as being a specifications for a system, subsystem, hardware or software etc. Therefore, the breakdown stops at a level where a BE can be realized by an assembly (hardware part) or a software package (software part). Assemblies and software packages can in turn have a defined Bill of Material or a defined assembly structure.

The advantage of using explicit breakdown structures is that all BEs and parts can be reused in many installation locations and in multiple breakdowns (eg for different products) without having to define new elements, and one can introduce new intermediate levels at any time during the project, without causing a lot of extra work.

Another advantage is that there is no need to define alternative BEs, since alternatives are just different part realizations for a BE.

The incorporation of breakdowns and parts from subcontractors is also simpler, because there is only a need to define a parent-child relationship between the main contractor’s leaf element and the subcontractors root element. This is much more cost effective than enforcing a complete identification system which the subcontractor needs to maintain.

### 5.3.3 Combining the two product breakdown approaches in S3000L

Even though the S3000L data model supports both approaches as such, a "combination" of the two approaches is the recommended way forward.

1. During the early phases of LSA:
   - Use the product breakdown, breakdown elements and their identification (BEI) as defined by design (including the identification of revisions in order to be able to identify and track design changes).
   - If there is a need to introduce BEs required by LSA, then:
     - Firstly, try to influence the breakdown structure as defined by design.
Secondly, if there is possibility to add this to the design, then create a separate branch in the breakdown. Do not enter intermediate BEs, since that will make it hard to manage updates of the breakdown coming from design

- If there is a need to add BEs to the design breakdown, incorporate product breakdowns from subcontractors using the subcontractor BEIs, if possible
- Let assemblies, subassemblies and components be identified by part numbers and not by its position(s) in a breakdown

2 At delivery of the LSA results

- Apply a systematic identification system to the BEs from the product itself down to a sufficient breakdown level (refer to Para 5.3.1). Proposal is to stop at the level where a breakdown element (specification) can be realized by a hardware or software part.
- Do not enforce BEI to parts, i.e. to actual realizations.

This approach will support both the flexibility required during the LSA process, and provide a cross-discipline solution.

5.4 Purpose of breakdown element identifier

The BEI should give a structured view of the product in relation to the defined breakdown. The following paragraphs provide a general overview of how these different breakdown methodologies can be documented.

The interrelationship of an item with other items at the same indenture level and the next higher indenture, as well as its breakdown items (eg attaching parts), will be reflected by the syntax of the BEI (traditional approach) or by parent-child relationship (PDM approach). To reflect the structure of an IUA assignment of hardware and software (SW), the following methods are recommended, depending on the requirements of the project.

- Pure functional breakdown
- Pure physical breakdown
- Mixed breakdown, functional and physical in one breakdown tree
- Pure functional and pure physical breakdown parallel, with cross-mapping

Special attention should be given to SW. In modern state-of-the-art systems, SW use is increasing. This should be considered in any logistic analysis. It should be possible to integrate loadable SW packages into the breakdown, if the installation/de-installation of SW is maintenance relevant. The assignment of SW to hardware should be clear within the breakdown. Configuration changes due to SW modification are another aspect of the breakdown. If hardware will not be modified, but SW integrated in the hardware will change, how can this situation be documented in a product breakdown.

5.4.1 Breakdown element identifier structure

For a traditional product breakdown approach, the BEI structure defines the relationship between the different breakdown indenture levels. The number and the type of characters assigned to each indenture level of the product breakdown are defined here. The BEI structure should be developed as early as possible and great care should be taken to define it appropriately to the project. However, BEIs should not be assigned to the BEs until the first delivery of LSA results (refer to Para 5.3.3). Avoid using more characters than required for each indenture level. Ensure that the BEI structure is capable, consistent, and broad enough to address the entire project structure. Know the limitations of your LSA application as it relates to BEI length and syntax. It is recommended to plan and allocate resources to address any system structure limitations.

Note

Projects using the traditional LSA product breakdown approach will notice that any changes to the BEI structure would cause rework for the different IT systems of the LSA and associated ILS disciplines. For these projects it is better to use an additional digit for an
indenture level precautionary than to encounter BEI structure problems at later stage of the project.

For a PDM product breakdown approach it is not required to define a specific BEI structure. In this case, the parent-child relations of the breakdown elements are realized by explicit relationship.

5.4.2 Alternative breakdown element
Projects using the traditional LSA product breakdown will meet the requirement to be able to define alternative BEs similar to the former usage of ALC from MIL-STD 1388-2B or Disassembly Code Variant in S1000D for example.

The requirement for alternative BEs can be the result of, for example, different manufacturers that are selected for an item at a specific installation place, but the respective manufactured item requires different logistic considerations (in form, fit, function and interface, they are the same). In S3000L, the usage of different BEI to document this situation is not mandatory. Alternate items to be installed at the same installation place within a product breakdown can be different realizations by different hardware (or even software) at the same installation place. The representation of these alternate realizations can be done in two different ways within S3000L:

- Introduction of additional BEs with own BEIs (and even possibly different revisions, BER) to document different development stages). Relation to different product variants can be realized by using product related applicability
- Introduction of different realizations of a BE by different realizations (eg from different manufacturers). These parts can become LSA candidates for the documentation of different logistic considerations, if required.

Note
For the documentation of alternative parts to be possibly installed at one and the same installation place, which are exactly the same in form, fit, function, interface and logistic considerations it is sufficient to document the alternate realization against the corresponding BEI without making the alternate part a LSA candidate.

Alternative BE is just another BE, and not a revision of an existing BE. Its BEI can indicate that it is alternative equipment or component to an existing BE, and the BEI therefore indicates that the BE can occupy the same installation position in the breakdown as an existing BE. This means that such an alternative BE will have its own unique BEI and its unique BER.

5.4.3 Documentation of peripheral items
Besides the product/system that is subjected to a LSA (prime LSA item), analysis of peripheral items may also be required. Those peripheral items (which include special tasks) have to be addressed by unique BEI and have their own product breakdown composed of eg BEs and parts. However it is necessary for the BEI to follow the BEI logic that is defined for the prime LSA item.

Examples of peripheral items (enabling products) are:
- LSA for complex support equipment (eg test equipment, working platforms)
- LSA for complex training equipment (eg simulators, training rigs)

5.4.4 Identification of item families
Similar items can be grouped together for analysis purposes. This family concept can minimize the efforts and can avoid the repetition of similar analysis activities. The following examples should help to find appropriate criteria for the use of a family BE represented by a corresponding BEI.

- Wiring of the same type (eg standard copper, coaxial, fiber-optical) within harnesses can be grouped together into families. Within these families, standardized tasks for cable maintenance can be identified
− Structural parts of the same type can be grouped together. Within these families, standard structure repair procedures can be defined.
− Grouping by manufacturer. All parts of a similar type coming from one manufacturer, can be grouped, because the maintenance of these parts is covered by a maintenance concept supported by the manufacturer itself (e.g. with corresponding repair kits).

5.4.5 Integration requirements
Integration is an important aspect within the logistics area. It should be ensured that each BE can be identified uniquely by each logistic discipline by both the customer and the contractor. It is recommended to consider the following aspects:

− Harmonization of terminology
  The item names (e.g. derived from early design documents) should be used unchanged and within the associated logistic disciplines in order to avoid confusion by the usage of different names for the same item.

− Harmonization of key addresses between the logistic disciplines
  The key addresses should be kept as common as possible in order to ease any communication or interaction in between different disciplines. For that reason, a common identification of BEs and parts shall be performed to a maximum extent within all logistic disciplines as well as the design department and CM. It is recommended to harmonize the numbering systems of LSA and other technical/logistic disciplines.

− Breakdown element and part identification adopted to a standardized numbering system
  The BEI should correspond to a SNS as used, for example, by the design group or by CM. It is recommended that the numbering systems of LSA and the design department and the CM should be harmonized.

5.4.6 Establishment of an exchange system for data and documents
By using the BEI and part numbers as a common key, the following requirements can be supported:

− Related ILS elements can be registered and exchanged internally and between partner companies
− Interfaces between the logistics disciplines, design and configuration can be established in order to establish an information network
− To ease the comparison of data between logistic disciplines in order to guarantee commonality and data quality
− To identify functional and/or physical positions of all LSA candidate items
− To ease trouble-shooting
− To address reports derived from or received by the LSA database (e.g. for logistic disciplines, management)
− To address data exchange with the customer, partner companies or suppliers
− To address LSA status information to breakdown elements
− To structure documents that are relevant for LSA and/or other disciplines (e.g. technical specifications, handbooks, general documents, drawings and document numbering systems)
− To enable structuring of other LSA documents such as the CIL

5.5 Breakdown element identifier logic
Within BEI logic, some special aspects of information can be incorporated. The main purpose of the BEI is the structuring of a product into hierarchical levels from the top level down to the required breakdown depth, in order to document the relevant equipment and components. The establishment of rules for the syntax of the BEI is important. These rules are agreed to at the LSA GC.
5.5.1 Breakdown without homogeneous BEI logic

For a complete documentation of breakdown methodology, the explicit parent-child principle should also be described. This means, that a breakdown can be established without having a structured BEI logic. Each BE only needs to identify its children. Each association with the respective child BE can contain information about the quantity of child elements (always defaulted to one) and possible description of installation location.

**Note**
Quantity and installation location is data being associated with the relationship between the parent and child BEs.

Table 5 Basic table for explicit breakdown documentation

<table>
<thead>
<tr>
<th>Item name</th>
<th>BEI</th>
<th>Part number</th>
<th>Item type</th>
<th>Next higher item</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product / end item</td>
<td>-</td>
<td>-</td>
<td>Product</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Fuel tank</td>
<td>TF001</td>
<td>-</td>
<td>Equipment</td>
<td>Product</td>
<td>3</td>
</tr>
<tr>
<td>Auxiliary tank</td>
<td>TF004</td>
<td>-</td>
<td>Equipment</td>
<td>Product</td>
<td>2</td>
</tr>
<tr>
<td>Rubber tank</td>
<td>TRT001</td>
<td>-</td>
<td>Component</td>
<td>TF001</td>
<td>1</td>
</tr>
<tr>
<td>Tank cover</td>
<td>TC001</td>
<td>-</td>
<td>Component</td>
<td>TF001</td>
<td>2</td>
</tr>
<tr>
<td>Blanking plug</td>
<td>TBP001</td>
<td>-</td>
<td>Component</td>
<td>TF001</td>
<td>1</td>
</tr>
<tr>
<td>Drain valve</td>
<td>TV001</td>
<td>-</td>
<td>Component</td>
<td>TF001</td>
<td>1</td>
</tr>
<tr>
<td>Valve1</td>
<td>-</td>
<td>45-90F</td>
<td>Component</td>
<td>TV001</td>
<td>1</td>
</tr>
<tr>
<td>Valve2</td>
<td>-</td>
<td>45-90H</td>
<td>Alternate component</td>
<td>TV001</td>
<td>1</td>
</tr>
<tr>
<td>Filter1</td>
<td>-</td>
<td>27-12</td>
<td>Component</td>
<td>45-90F</td>
<td>1</td>
</tr>
<tr>
<td>Body</td>
<td>-</td>
<td>45-BB</td>
<td>Component</td>
<td>45-90F</td>
<td>1</td>
</tr>
<tr>
<td>Filter2</td>
<td>-</td>
<td>27-27</td>
<td>Component</td>
<td>45-90H</td>
<td>1</td>
</tr>
</tbody>
</table>

The breakdown structure example from Table 5 is visualized in Fig 15.

**Note**
Normally, any functional information is missing in this type of explicit breakdown methodology.
5.5.2 Basic syntax of breakdown element identifier – Establishment of breakdown levels

The implementation of a BEI syntax must consider the definition of the different indenture levels and how to make the BEI easily comprehensible. Both the number of digits per indenture level and the proper usage of letters, numbers and special characters must be defined carefully for each project. Each aspect concerning the required depth and range, readability and possible future requirements must be considered. Readability can be improved by taking into account the following aspects:

- The usage of separators between different levels is recommended
- A decision made on whether to always use the full range of digits in the breakdown syntax or to allow a BEI termination at the relevant indenture level.

<table>
<thead>
<tr>
<th>BEI</th>
<th>BEI (for better clarity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>28000000000AAA</td>
<td>better readable by usage of a separating character</td>
</tr>
<tr>
<td>28000000A</td>
<td>better readable by usage of a separating blank</td>
</tr>
<tr>
<td>28-000-000-000-000</td>
<td></td>
</tr>
<tr>
<td>28-001-000-000-000</td>
<td></td>
</tr>
<tr>
<td>28-001-001-000-000</td>
<td>termination after the last relevant breakdown level</td>
</tr>
<tr>
<td>28-001-001-002-000</td>
<td></td>
</tr>
<tr>
<td>28-001-001-002-001</td>
<td></td>
</tr>
</tbody>
</table>
In Fig 16 a typical example for a simple BEI syntax is given:

```
28-01-01-001-001
```

- **BEI, 5th indenture level**, 3 digits, numeric
- **BEI, 4th indenture level**, 3 digits, numeric
- **BEI, 3rd indenture level**, 2 digits, numeric
- **BEI, 2nd indenture level**, 2 digits, numeric
- **BEI, 1st indenture level**, 2 digits, numeric

**ICN-B6865-S3000L0016-001-001-A**

**Fig 16 Example for simple BEI syntax (reduced to numbering system)**

In addition to readability aspects, other information can be incorporated within the BEI. Examples are:
- **BE variant coding**
- **Coding of the project/product**, eg a product identifier
- **BEI suffix**, eg part of coding to document BE variants
- **BEI prefix**, eg product/project identifier

**ICN-B6865-S3000L0016-001-001-A**

**Fig 17 Example for extended BEI syntax**

**Note**

The definition of the syntax of a BEI as a code is not restricted. The BEI can be used as a simple representation of the numbering system as shown in Fig 16. To incorporate additional information, the BEI can be extended eg by a prefix or a suffix as shown in Fig 17. In this way the documentation of different BE variants can be realized. JX-28-01-01-001-001-A and JX-28-01-01-001-001-B would be two different variants of breakdown elements being installed at the same installation place. The installation place is documented by the part of the BEI representing the numbering system of the product breakdown, the variant is identified by the suffix.

### 5.6 Breakdown element identifier and related codes

Along with the coding of the functional/physical breakdown element with the help of a BEI, additional information is important for the unique identification of any product component. The usage of this information described in the following paragraphs along with the BEI will provide a complete idea of the structure and assembly of the analyzed product.

#### 5.6.1 Purpose of breakdown element revision

The Breakdown Element Revision (BER) documents different stages of development. Especially in the design and development phase it can be necessary to document different development step concerning influence on design and principal changes which have significant influence on technical/logistic analysis results and/or on the support solutions. Revisions of BE
have not been considered in traditional product breakdown methodologies, but is being introduced in S3000L in order to enable a better integration with product design or logistic disciplines.

Note
The BER is not designated to be the code to distinguish between for example, alternate hardware components (eg from different manufacturers) which can be installed at the same installation place (also refer to Para 5.4.2).

5.6.2 Applicability
Often different equipments (eg from different manufacturers) or equipment variants (eg from the same manufacturer, but different releases) can be installed at the same installation position within a product. The usage of applicability for different BEs and BE realizations (eg parts) enables for the distinction between variants of equipments/components documented by BEI or part identifier which can be installed within different variants of products (documented by applicability). Concerning variants of components or parts respectively it is possible to distinguish between 2 different situations:

− Situation 1:
  Simple alternative components (equal in form, fit, function, interface and logistics considerations, eg interchangeable components from different manufacturers).

− Situation 2:
  Modified alternative components (equal in form, fit, function and interface, but not equal concerning logistics, eg equipment/parts of the same type from different manufacturers, which have different maintenance concepts).

In general, situation 2 has to be documented carefully by the introduction of different elements within the product breakdown. These elements can be different BEs represented by different BEI (traditional product breakdown approach) or, alternative, different realizations by eg different parts (PDM breakdown approach) of one and the same BE. In both cases it is possible to document the different logistic information against the LSA candidate, which is represented in the first case by the BEI and in the second case by the BE realization (installation of different parts at the same installation place, eg identified by part number).

Additional to the requirement to be able to distinguish between variants of components or parts respectively, the product as the superior item can also be realized by different product variants (eg for different customers). A proper applicability method must be able to allocate each component or part variant to the corresponding product variant.

In the following S3000L product breakdown example the principle of assigning applicability is visualized. The breakdown is represented by PDM-orientated explicit parent-child relationships and applicability is realized by relationship to the product variant applicability object within the product breakdown. The same principle can be used when a traditional product breakdown approach is realized.

Note
In comparison to the traditional approach by MIL-STD 1388-2B the function of the data elements ALC and UOC is now realized by applicability and hardware or software realization.
As shown in Fig 18, two different variants of the product, in this case a fuel truck, are contracted. The main difference is the installation of two different engines. For the truck variant represented by the BEI A00, a gas engine is installed, for the truck variant represented by the BEI A01, a diesel engine is installed. The breakdown elements Body and Transmission are applicable for both truck variants. The usage of applicability represented by Fig 18 easily enables a complete representation of each product variant.

For the representation of different variants of component/part, which can be installed on an equipment (represented by one BE), applicability is realized by hardware (or software) element realization. The three different fuel pumps which can be installed into the gas engine are variants of the fuel pump (represented by the BE fuel pump) with the BEI A20300.

**Fig 18 Usage of BEI and applicability**

**Table 7 Product variant applicability from Fig 18**

<table>
<thead>
<tr>
<th>BEI</th>
<th>PNR</th>
<th>Name</th>
<th>Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A00</td>
<td>R01</td>
<td>Truck (truck model A)</td>
<td>applicability object</td>
</tr>
<tr>
<td>A01</td>
<td>R02</td>
<td>Truck (truck model B)</td>
<td>applicability object</td>
</tr>
<tr>
<td>A100</td>
<td>BY02</td>
<td>Body</td>
<td>to A00, A01</td>
</tr>
<tr>
<td>A200</td>
<td>GE001</td>
<td>Gas engine</td>
<td>to A00</td>
</tr>
<tr>
<td>A201</td>
<td>DE001</td>
<td>Diesel engine</td>
<td>to A01</td>
</tr>
<tr>
<td>A300</td>
<td>TT01</td>
<td>Transmission</td>
<td>to A00, A01</td>
</tr>
</tbody>
</table>
It is strongly recommended that the identification of variants (both, product/end item and its components/parts) and the usage of applicability between contractor and customer is clarified at the LSA GC.

5.6.3 **Breakdown element type**

BEs can be of both functional and physical character within product breakdowns. To distinguish functional and physical BEs a type information can be used, especially if a mixed methodology will be applied. The classification of BEs, representing a product variant or an assembly, can also be covered in this way. The type information can be introduced as an attribute of the BE itself within the LSA database.

5.6.4 **Equipment type**

Other important information concerning equipment which can be assigned to the BE relate to the installation and repair of the item. This includes discard information as well as accessibility information. In the following tables, a number of general terms are introduced. This table should be a guide to how corresponding terminology can be used within a project. However, it is recommended that a common understanding of typical logistic terms eg within a glossary, is agreed at the LSA GC.

**Table 8  Aspect of replaceability**

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct replaceable</td>
<td>Item that is directly replaceable on the product level (a LRU as already described in Para 5.1.6 is a typical direct replaceable item)</td>
</tr>
<tr>
<td>Not direct replaceable</td>
<td>Item that is not directly replaceable on the product level (installed in a superior item). The superior item must be removed before replacement, eg the compressor of an engine can only be replaced after engine removal (a SRU as described in Para 5.1.6 is a typical “not direct replaceable” item)</td>
</tr>
<tr>
<td>Non replaceable</td>
<td>Item that is intrinsically tied to another and cannot be replaced separately.</td>
</tr>
</tbody>
</table>

Additional to the replaceability it is necessary to know whether an equipment or a component can be repaired or not. This aspect is summarized in the following table.

**Table 9  Aspects of repairability**

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full repairable</td>
<td>Item that is fully repairable</td>
</tr>
<tr>
<td>Partial repairable</td>
<td>Item that is partial repairable (depends on failure)</td>
</tr>
<tr>
<td>Non repairable</td>
<td>Item that is not repairable.</td>
</tr>
<tr>
<td>(discard part)</td>
<td></td>
</tr>
</tbody>
</table>

Normally, the repairability information is related to a component/part and will be documented against a realization (eg part). In some cases it can be an attribute of a BE, when the installation situation prevents a repair of the installed component.
The combination of replaceability and repairability fully describe the equipment within the context of the decision on whether an item will be replaced, but this decision can also be influenced by economical considerations.

5.7 Breakdown element identifier requirements concerning software
Software as a part of the breakdown has some special requirements. In general, it should be possible to include SW items within the overall product breakdown. Refer to Para 11.3 for examples concerning SW elements in physical and functional breakdowns.

5.8 Breakdown element identifier aspects for LSA guidance conference
To summarize the most important aspects, the following list supports the creation of rules for the establishment of a product breakdown methodology.

5.8.1 General aspects for customer involvement
The following aspects concerning breakdown depth must be considered:

- Have overall support and repair strategies with influence on breakdown requirements already been decided? (eg two level maintenance concept, overall test/BITE architecture, single source principle concerning major repair)
- What aspects must be observed concerning the depth of detailed breakdown?
  - Potential depth of repair at authorized level (eg at depot level)
  - Potential level for traceable usage data collection
- Have rules been established to differentiate between breakdown requirements concerning:
  - Newly developed or major modified equipment
  - Moderate modified equipment
  - Commercial Off-The-Shelf (COTS) items
- Have rules been established for SW items concerning:
  - Equipment with embedded SW (breakdown concerning SW design aspects)
  - Equipment with user loadable SW and data (SW- up/down loading, transportation)
- Is a dedicated breakdown concerning SW items predicted (rules to be followed)?
- Are peripheral items (eg complex support equipment, simulators, and maintenance trainers) to be included in the breakdown?
- Are contractual requirements identified that need to be verified against a BE?
- For what program phases are LSA to be performed (especially starting stage)?
  - Pre definition or definition phase
  - Early development phase (eg breakdown for comparison purposes)
- Information to be addressed by the BE (eg required within CIL):
  - Marker indicating each selected LSA candidate
  - Marker indicating (potentially) cost-/readiness-/maintenance drivers
  - Marker for LRUs
  - Marker for SRUs
  - MTBF values allocated to each BE or part
  - Estimated unit price (or price categories) for high cost articles – as available

5.8.2 Standardized numbering system
A numbering system to be used for the product breakdown has to be identified, eg SNS from S1000D to be used for BEI structure.

Have rules been established concerning the breakdowns type to be defined/used?

- Physical breakdown
− Functional breakdown  
− Mixture of functional and physical breakdown  
− Both types in parallel linked by cross-mapping

Use of the product breakdown for standardization purposes:

− Is the selected BEI to be used as central identifier for assigned LSA candidates within a CIL  
− Is the selected BEI to be used for item terminology (common for all logistic disciplines)

6 Candidate item selection and identification

The LSA process should be seen as considerably cost intensive. For that reason, the selection of the LSA candidates and the associated analysis activities should be done with adequate care in order to keep a good balance between the effort for LSA and the benefits acquired from the LSA process. The candidates can be of different quality, which requires the type of analysis processes and the depth of the analysis to be dependent on the particular LSA candidate.

6.1 Definitions and general terms

First, some definitions of general terms used in the logistics environment concerning systems/items impacted by logistic support analysis will be introduced.

6.1.1 Candidate and non-candidate

The LSA candidate is the driver for all LSA activities. A potential LSA candidate, in general, can be an item on system, subsystem, equipment, module or sub module level, which is repairable or which requires maintenance support, scheduled or unscheduled. The decision on whether an IUA will be part of the CIL is dependent upon the criteria above, plus some additional criteria which must be defined individually by the project.

In addition to this hardware related definition, special activities or descriptions can be part of a product breakdown. For example, the description of standard practices for the repair of structural components can be documented within a special chapter of the product breakdown. In this case, these specific non-hardware breakdown elements also can be LSA candidates, because, for example, special events or maintenance tasks are addressed to them.

For non-candidate items, a detailed logistic analysis is not required. These items do not need to appear in a physical breakdown under a unique BEI. If a deep breakdown is established, all items that do not fit into any of the selecting criteria given in the LSA candidate selection checklist are potential non-candidates. Breakdown items that are only subject to standard tasks for which no special logistic resource is required, are also potential non-candidates. Typical examples for non-candidates are consumables and bulk items, such as screws, bolts, nuts or washers. Nevertheless, even though an item may be identified as a non-candidate during the LSA candidate selection process, the result needs to be documented in the CIL.

6.1.2 Maintenance relevant items and maintenance significant items

For a clear definition of existing and new terms, a separation between the established term Maintenance Significant Item (MSI) and the definition of the LSA candidate must be established. Any IUA that is impacted by any maintenance action must be regarded as relevant for logistic analysis. The type of maintenance tasks is divided into two general categories: the scheduled or preventive maintenance and the unscheduled maintenance (failure or damage correcting). For this reason, the terms will be defined as follows:

<table>
<thead>
<tr>
<th>Table 10 Definitions of MSI and MRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item type</td>
</tr>
<tr>
<td>Maintenance Relevant Item</td>
</tr>
</tbody>
</table>
**Item type** | **Definition**
--- | ---
Maintenance Significant Item | A Maintenance Significant Item is an item that was identified by any selection process as the result of a SMA procedure such as S4000M, MSG-3 or Reliability Centered Maintenance (RCM). For this item, a SMA will be performed (e.g., a system and power plant analysis as described in S4000M). A MSI can become a LSA candidate if the SMA identifies a scheduled task. This task will be documented in the LSA database.

### 6.1.3 Structural items, structure significant items and structural details

The structure of a system (like the bodywork of a car) is also part of the product breakdown. Structural items can be typical LSA candidates and also candidates for a SMA.

**Table 11 Definitions of Structural Item, Structure Significant Item and Structural Detail**

<table>
<thead>
<tr>
<th>Item type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Item</td>
<td>A Structural Item is a part of the system's bodywork. It can be a LSA candidate as well as a Structure Significant Item.</td>
</tr>
<tr>
<td>Structure Significant Item</td>
<td>A Structure Significant Item is an item which was identified by any selection process from a SMA procedure such as MSG-3 or S4000M. For this item, a SMA will be performed (e.g., Structure Analysis as described in S4000M).</td>
</tr>
<tr>
<td>Structural Detail</td>
<td>A Structural Detail is a specific area of an SSI which was identified by any selection process from a SMA procedure such as MSG-3 or S4000M. For this detail, a SMA will be performed. In the breakdown, a structural detail can be identified by an additional artificial BEI below the superordinate structural component.</td>
</tr>
</tbody>
</table>

**Note**

A detailed classification of structural items (with impact on a product breakdown) and a definition of zones are given in S4000M for the application of the SMA.

### 6.1.4 Non-hardware items

Within a hierarchical hardware breakdown, only hardware is normally documented. If it is necessary to describe any general tasks which cannot be assigned to a specific hardware or which are standard procedures for a group of items (e.g., repair procedures for electrical wiring), non-hardware items must exist within the product breakdown. These non-hardware items can become LSA candidates, too.

### 6.2 Classification of LSA candidates

It is recommended to establish categories of LSA candidates. This enables the differentiation between the needs and to reduce the effort to an adequate level for each category of LSA candidate. The LSA candidates can be classified according to the table below. For all candidate types, the exact meaning of every classification must be defined carefully and harmonized with the customer, especially for the partial candidates, as it may be necessary to define more than one partial type within a certain project.

**Table 12 LSA candidate categories**

<table>
<thead>
<tr>
<th>Candidate category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full candidate</td>
<td>Provide full scale of selected LSA information applicable to the related item.</td>
</tr>
</tbody>
</table>
**Candidate category** | **Description**
--- | ---
Partial candidate | Provide a partial scale of selected LSA information applicable to the related item, (eg only remove and install information because of the need to gain access to other items, but no repair information required because the item is a discard item).

Candidate family | Provide LSA information focused on specific requirements, (eg for a family of items such as non-specific wirings, harnesses, pipes, lines, minor items of the same type such as clamps, harnesses, connectors) Those items may need only a summary analysis for each group.

Standard procedures candidate | Provide a partial scale of LSA information because relevant information for this item is already covered by the LSA information of other non-hardware BEI (eg tasks concerning standard repair procedures of structure or electrical wiring).

For every LSA candidate type, a list of criteria must be established. Together with these criteria lists, a flowchart for the decision process should be developed to support the logisticians.

### 6.3 Selection process and criteria list

For LSA candidate selection, each breakdown item must be assessed concerning its significance for LSA purposes. If the agreed upon section matrix indicates that an item is a candidate according to the rules for selection, the relevant item will be noted as a LSA candidate.

The LSA candidate selection process should be adapted to each project. In the following paragraphs, examples for typical selection criteria are given. Additionally, LSA candidate selection flowcharts are given in Para 11.4. Depending on the project, some questions in the flowcharts can be modified or additional questions can appear in the process.

To perform the LSA candidate selection process, some preconditions concerning product breakdown and the establishment of rules must be fulfilled.

#### 6.3.1 Preconditions for the selection process - Product breakdown

LSA candidates will be derived from a hierarchical hardware breakdown of the System under Analysis (SUA) from the product level to sub-module level. For this reason, the most important precondition to be able to perform a LSA candidate selection is the availability of a product breakdown. It has to be ensured, that the existing breakdown is applicable for this task by examination of the following criteria:

- Is the product breakdown available in a sufficient depth and extent (refer to Para 5)?
- Is the product breakdown a preliminary issue (eg for responding to a request by a potential customer or for pre-contractual agreements with a potential customer)?
- Is the existing product breakdown already agreed by the customer?

#### 6.3.2 Preconditions for the selection process – Existing analysis results

Each existing analysis result (eg from analysis activities in advance or directly from the manufacturer) should be used for candidate selection. Examples are the following:

- Existing experience for equipments already used in other products/systems
- Existing maintainability or reliability data
- Existing Failure Mode and Effects Analysis (FMEA) or Failure Mode and Effects Criticality Analysis (FMECA) results
6.3.3 Preconditions for the selection process - Candidate selection rules

The establishment of the rules for LSA candidate selection and the extent of analysis must be agreed to by the contractor and the customer as a part of the LSA GC. This means that the final CIL cannot be presented at the beginning of a LSA GC (only a preliminary CIL can be published). In the following list, some criteria are listed that can be used as a basic guideline for potentially relevant aspects for a LSA candidate selection.

The following items should be considered as potential LSA candidates that need to be assessed for inherent reasons and/or due to their installation area:

- The item is maintenance significant concerning the related failure/defect frequency or the awaited workload.
- The item needs special logistic requirements such as personnel, training, material or support equipment.
- The item is subject to scheduled or preventive maintenance (e.g., preventive change of life-limited items).
- The item is subject to special procedures, e.g., after special events like lightning-, bird-, hail-strikes, contact with obstacles.
- The item is subject to diagnostic and/or functional test tasks (e.g., complete system tests linked to a high level BEI).
- The item is potentially endangered by damages due to the installation area and/or the design of an item.
- The item is subject to the use of new technology
- The item contains user loadable software (including data)

The following items should be considered as a potential LSA candidate that needs to be assessed for a global point of view (for special interests):

- The item is a potential readiness driver
- The item is a potential cost driver (e.g., expensive support equipment required)
- The item is a potential maintenance driver (e.g., high workload expected)
- The item is subject to emphatically customer interest
- The item is subject to LSA related contractual fulfillment

The following items should be considered as potential LSA candidates that need to be assessed only or mainly for standardization reasons:

- The items require only removal and installation in order to gain and undo access to a LSA candidate. No true LSA activities are required concerning these items, but the involved logistic requirements should be standardized and documented.
- Documentation of general tasks within the LSA database. Those tasks need to be incorporated in the LSA database mainly for registration and documentation of the associated logistic requirements. Often, the general tasks are linked to non-hardware BEI.
- Groups of items that may require a summarized logistic analysis (e.g., a family of non-specific wirings, harnesses, pipes, lines, minor items of the same type like clamps)

All the criteria above must be detailed using measurable threshold values. For example, the criteria indicating whether an item is maintenance significant concerning the related failure/defect frequency, must be specified with a clear threshold value such as the MTBF. The criteria above fit for all types of candidates. The possible criteria for the differentiation of the candidate types will be described in the following paragraph.

6.3.4 Preconditions for the selection process - Candidate classification selection rules

In the following tables, examples for criteria for the different candidate types are given to support this important task of the LSA candidate selection process. Flowcharts for candidate item selection are given in Para 11.4.
Table 13  Classification criteria for full LSA candidates

<table>
<thead>
<tr>
<th>Candidate classification</th>
<th>Classification criteria</th>
<th>Required answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full candidate</td>
<td>Is the item a newly developed or a major modified item (functional equipments, not valid for structure)?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is the item a Line Replaceable Unit (LRU)?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is the item a repairable item?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is the item of low reliability (must be defined in measurable way)?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is the item maintenance relevant? In other words, are there any dedicated maintenance tasks planned such as repair, servicing, lubrication or calibration (no general tasks such as simple cleaning or a simple replacement).</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Are dedicated maintenance tasks complex, very time consuming or personnel intensive?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is the required support equipment for dedicated maintenance tasks a non-standard or a non-existent item?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Does the item need specific scheduled or preventive maintenance identified in a Scheduled Maintenance Analysis (SMA)?</td>
<td>Yes</td>
</tr>
</tbody>
</table>

If all of the questions from Table 13 are answered with "No", the IUA will not be a full candidate. If any question is answered with "Yes", then this item is a potential full candidate and can be classified as such in the CIL.

Table 14  Classification criteria for partial LSA candidates

<table>
<thead>
<tr>
<th>Candidate classification</th>
<th>Classification criteria</th>
<th>Required answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial candidate</td>
<td>Must the item be removed to gain access?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is a removal for access frequent for the IUA?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is the item a system or a subsystem that has not been previously defined as a full candidate, for which a fault location and/or a test procedure or other general maintenance procedures will be described?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is the item a non-hardware item, for which general activities (eg cleaning, storing, parking, mooring, general inspections) will be described?</td>
<td>Yes</td>
</tr>
</tbody>
</table>

If all of the questions from Table 14 are answered with "No", the IUA will not be a partial candidate. If any question is answered with "Yes", then this item is a potential partial candidate and can be classified as such in the CIL.
Table 15 Classification criteria for LSA candidate family

<table>
<thead>
<tr>
<th>Candidate classification</th>
<th>Classification criteria</th>
<th>Required answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate family</td>
<td>Is the IUA installed within the system many times in the same or similar way?</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Is it possible to combine all equal or similar items to one family with common maintenance activities?</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The LSA candidate family concept should be used when a large number of equal or similar items are installed within the SUA. For example, all electrical wiring with the same properties and the same or similar connectors can be summarized under one LSA candidate family. For this family, all maintenance relevant information (e.g., a connector repair concept) is valid for all single cables of the whole family, and can be documented for a family BEI.

6.4 Influencing factors

The following factors should be taken into account for candidate item selection and/or establishing related rules:

- Program phase to which the LSA program belongs
- Complexity of the product/system to be analyzed
- Price of the product/system to be analyzed
- Budget available for the overall logistic support and budget specifically planned for the LSA process. The limitation of budget will always have an influence on the number of LSA candidates that can be analyzed. In these cases, a reduction in the real maintenance and cost drivers is required.
- Importance of the LSA results for the internal logistic disciplines and the information required by the customer and the industry management required to make decisions and to fulfillment the contractual requirements of the LSA related items.
- Items already in use under comparable conditions
  - Items already in use, but under not comparable conditions
  - COTS items usable with or without major modification
  - Items already available but requiring major modification
  - Newly developed items
- Information expected from the LSA process (to be harmonized with/agreed by the customer during the LSA GC)
- Identification of possible alternatives (for hardware, software, support concepts) and related consequences (e.g., MC, logistic support requirements)
- Proposal of recommended MC
- Identification of tasks associated with the intended MC
- Identification of related logistic support requirements
- Estimation of related logistic support costs (part of LCC)

6.5 Recommendations for LSA candidate selection

To give an overview to the analyst on how to handle LSA candidate selection criteria in detail, a list of recommendations concerning what is unquestionably required and what is nice to have. The following table provides assistance in the selection process. From this table, the analyst can find recommendations on how to determine whether it is mandatory, recommended, or voluntary for an IUA to be a LSA candidate.
Table 16  Recommendations for LSA candidate selection

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Description of criteria</th>
<th>Consideration as potential candidate item?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk and criticality</td>
<td>Must LSA candidates be integrated from separate LSA programs? LSA information to be integrated into one LSA program (developed outside) without any change (eg LSA data for an engine derived from a separate contractor) or external LSA information to be integrated with adoption to a LSA program that requires changes to be harmonized with the original contractor. The IUA is subject to a SMA (S4000M, MSG-3, RCM) and scheduled or preventive tasks are identified. The structural IUA is subject to a SMA (S4000M, MSG-3, RCM) and scheduled or preventive tasks are identified. The IUA is subject to life limits. The IUA is subject to preventive maintenance, other than scheduled, or to special procedures (eg after special events like lightning-, bird-, hail- strikes, contact with obstacles). The IUA is subject to the use of new technologies. The IUA is a potential cost driver (high value items, cost limits must be defined with the customer). The IUA is a potential readiness driver because of long repair times. The IUA is a potential maintenance driver (high workload). The IUA is of special customer interest (information values must be defined with the customer) The IUA is subject to LSA related contractual fulfillment.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Item type</td>
<td>Item already in use under comparable conditions Item already in use under non-comparable conditions Item already in use but requiring major modification Item newly developed COTS items Part of an maintenance significant &quot;item family&quot; Item containing user loadable SW and/or data</td>
<td>Recommended, Mandatory, Recommended, Mandatory, Recommended, Recommended</td>
</tr>
<tr>
<td>General attributes</td>
<td>Item is line replaceable Item is shop replaceable Item is line repairable Item is shop repairable</td>
<td>Mandatory, Recommended, Recommended, Recommended</td>
</tr>
<tr>
<td>Aspects</td>
<td>Description of criteria</td>
<td>Consideration as potential candidate item?</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Maintenance significance</td>
<td>Item is repairable at customer depot or at industry level</td>
<td>Voluntary</td>
</tr>
<tr>
<td></td>
<td>A FMEA/FMECA is available concerning the related item</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td>Item is potentially subject to LSA relevant servicing</td>
<td>Voluntary</td>
</tr>
<tr>
<td></td>
<td>Item is potentially subject to (diagnostic/functional) test</td>
<td>Recommended</td>
</tr>
<tr>
<td></td>
<td>Item is potentially endangered by damages</td>
<td>Voluntary</td>
</tr>
<tr>
<td></td>
<td>Item is potentially subject to general tasks</td>
<td>Voluntary</td>
</tr>
<tr>
<td></td>
<td>Item is potentially subject to summary analyses</td>
<td>Recommended</td>
</tr>
<tr>
<td></td>
<td>Item is potentially subject to standard procedures</td>
<td>Voluntary</td>
</tr>
<tr>
<td></td>
<td>Item is subject only to gain/undo access to LSA candidates</td>
<td>Voluntary</td>
</tr>
</tbody>
</table>

### 6.6 Candidate item list (draft)

For LSA candidate selection purposes, a matrix may be used in order to structure and document the selection results. A first draft should be included in a recommendation list, prepared by the contractor, which reflects the proposed selection criteria. The draft of the CIL should be prepared by the contractor for the LSA GC. After harmonizing the proposed selection rules between the contractor and the customer, this draft can be reworked and published as the preliminary CIL.

### 6.7 Candidate item list as an output of LSA guidance conference

The CIL should be included in an obligatory list, reflecting the decisions that were made by the customer. This CIL must be a living document in order to document changes during the life cycles of the program. During the LSA GC (refer to Para 4) the Draft CIL will be discussed and harmonized between the customer and the contractor. This must be recognized as one of the most important steps regarding contractual terms. In the CIL, a collection of all LSA candidates and the corresponding workload from the selected analysis activities will be managed. To support the LSA process management, the progress of the LSA and documentation work should be documented by status codes, which reflect the status for all the different contractual relevant analysis activities to be covered within the LSA process.

To increase effectiveness, it is recommended to integrate the CIL as a central management tool into the logistic database application so that it will be possible to hold all relevant information of the CIL in the breakdown of the system. An overview of the project can easily be given by the corresponding reports.

All changes in the CIL for whatever reasons (eg technical, budgetary, redesign) must be discussed and harmonized between the customer and the contractor and documented carefully. For the LSA managers and the ILS managers for both the customer and the contractor, the CIL is the valid document for control of the whole LSA process. The customer and the contractor must understand, at every stage of the project, who is the holder of the contractually valid issue of the CIL.

### 7 Logistic analysis activities

In the early life cycle, LSA shall support cost relevant decisions by performance of adequate front end analysis (eg LORA). LSA must influence design concerning logistic aspects from the beginning of the definition phase and this task has to be continued during the lifetime of the
product. The inputs for the design come from different analysis sources such as reliability, maintainability, testability or SMA. The performance of analysis activities for LSA candidates can be a cost concern if no threshold is established to reduce the effort to a suitable level. In order to balance between analysis-related efforts and acquired knowledge which is mandatory to identify adequate logistic requirements (that satisfies the user's needs and logistic support cost constraints), a series of assessments is required. Related allocation criteria and rules must be identified by the contractor and proposed to the customer as general guidelines tailored to meet the specific requirements of a given program.

**Note**
This has a large cost influence on both the effort to be spent for the LSA program itself and on the generation of future support costs during the overall life cycle of the items under analysis. Based on experience acquired within a broad range of LSA applications, it is strongly recommended to perform this task on a preliminary basis as an initial assessment early in a program and to harmonize the derived strategy with the customer before signing a contract.

### 7.1 Principles for analysis activity selection

Only those analysis activities should be selected for a LSA candidate that are considered to improve the investigation result such a way that the expected benefit is greater than the effort to be spent for the intended analysis activity itself. Prior to the allocation of potential analysis activities to LSA candidates, they should be grouped into categories in order to identify the real need for analysis activities to reduce the effort to an adequate level (refer to Para 5).

Knowledge acquired by "lessons learned" should be taken into account on customer and contractor side. This holds for both, positive and negative experience. Based on that knowledge, LSA relevant decisions may be predetermined without further analysis activities.

New available analysis methods should be taken into account where feasible. For example, a simulation result is much more comprehensive than a traditional ("more static") analysis.

**Note**
Based on the experience of Industry, powerful simulation software packages are available and are expected to become important for LSA purposes.

### 7.2 Potential analysis activities

The following LSA activities are a list of potential analysis which can be performed and documented in different ways depending on the kind of project. The LSA database must document the LSA activities and their results.

- Analysis for identification of general LSA Needs
- Comparative analysis
- Human factor analysis
- Configuration assessment
- Reliability analysis assessment
- Maintainability analysis
- Testability analysis
- LSA FMEA - Logistic FMEA
- Damage analysis
- Special event analysis
- SMA (S4000M, MSG-3, RCM)
- LORA
- MTA
- SSA
- Operations analysis
- Simulation operational scenarios
Training Needs Analysis (TNA)

For each LSA activity, the contractor and the customer must agree whether, and how, the results of each analysis activity will be documented, delivered, commented and assessed. It is recommended that most of the results of the analysis activities be documented in a LSA database, but it is possible to document as a special analysis report or an official document. It must be clarified before the beginning of the analysis, the required results and the benefits from the collected data, and how the data will be evaluated and used for later logistic purposes.

7.2.1 Analysis for identification of general LSA needs

This starting point for any LSA activity is the basic precondition for applying any of the following tasks. Here you have to determine why the analysis must be performed. Do not spend valuable resources for analysis work without a clearly defined objective. The identification of general LSA needs is mainly related to contractual, customer and user interests. The following aspects must be considered and harmonized between the contractor and the customer very carefully to avoid any misunderstandings in a later stage of the project.

- Consider the results of the establishment of product usage data (ORD and CRD), intended support strategy and principles and alternative solutions. Clarify which scenarios must be analyzed, how and in what depth.
- Consider which contractual information must be demonstrated/validated by LSA results, how and in what depth.
- Consider the definition of required reports concerning areas of special interest (e.g., management reports, status overviews, performance measuring reports, verification reports).
- Also consider trade-off analysis results in order to guarantee the best return of investment. This means, for example, customer demands may be discussed to achieve the goal by some other method. The alternative solution might have significantly less effort, but it must lead to a comparative result which meets the requirements of the customer in a similar and acceptable level.

The result of these first considerations must be an agreement between the contractor and the customer about the purpose, goal, depth of analysis, and documentation method for each analysis type so each has a common view on the performance of LSA. The result of this analysis should be documented in the LSA program plan at the LSA GC.

7.2.2 Comparative analysis

This analysis will be performed upon special request only. As a precondition, comparative information of equipment, a system or an end item must be available and effective for decision making (limited to the initial LSA activities). If a comparative system can be defined, appropriate analyses must be performed in order to derive applicable comparative data. For the documentation of the results, special summary reports concerning comparative factors and data and/or support alternatives addressed to comparable LSA candidates must be identified.

7.2.3 Human factor analysis

This analysis may have influence on those LSA candidates for which a MTA is performed. Human factors can be a reason for special limitations (e.g., concerning the applicability of a maintenance task which requires special abilities). Ergonomic aspects must be considered here, as well as the definition of rules for an appropriate man-machine interface. For example, some aspects that can have limiting consequences to LSA activities (e.g., MTA):

- Ability to lift and carry heavy loads
- Ability to move for a long time in special conditions
- Handling of dangerous materials
- Limitations of personnel employment by law (security restrictions)
7.2.4 Configuration assessment
In the case of significant deviating design configurations of any LSA candidate, configuration assessments are mandatory. Potential change of validity, engineering changes, deviations or waivers must be examined. Configuration assessments must be performed as general assessments of the need of deviating maintenance concepts and/or logistic support requirements (personnel, material) in order to re-assess the need for analysis activities on LSA candidates.

The customer and the contractor must clarify how configuration deviations of LSA candidates should be documented within a logistic database. For that purpose, the usage of alternate BEI concepts is described in Para 5.

7.2.5 Assessment of reliability analysis
Reliability predictions (e.g., Mean Time between Failures, MTBF) related to the intended usage scenario of the LSA candidates must be assessed regarding their use for LSA purposes. The determination of reliability values is a design and development related task, but the documentation of certain results of this analysis is of crucial importance for LSA purposes. Normally, a reliability analysis is mandatory for each hardware LSA candidate. Additionally, it is recommended to extend reliability predictions to all essential items of a LSA breakdown and to ensure consistency of these values over all breakdown cascades.

Note
The reliability values are directly linked to the values of a LSA FMEA concerning ratios of different failures of a LSA candidate. These two areas in a logistic database must be consistent. Any LSA software package should be able to check this correlation between these two analysis areas automatically, otherwise the overview of reliability data consistency with FMECA data will be easily lost. Every LSA software package must be able to automatically check the consistency of reliability values over all breakdown cascades.

7.2.6 Maintainability analysis
For each LSA candidate, a maintainability analysis is strongly recommended in order to decide, from a technical point of view, whether an item is supportable or not (rules for related conditions and time frames should be established). Alternative repair solutions must be investigated, if applicable. Where applicable, a maintainability analysis must be applied that addresses the following aspects:

− Assessment of maintainability features reflecting customer requirements
− Identification of maintenance tasks at different levels in order to support the maintenance concept for each applicable LSA candidate
− Investigation of supportability aspects such as modular design of end item, good accessibility, installation concept in special zones (the item with the worst MTBF must not be installed behind others)

In general, the maintainability analysis can be performed by following applications with different level of detail and effort:

− Best Engineering Judgment in the case of lowest level of available information
− Assessment with or without transformation of equipment supplier information. In the case of demanding information from suppliers, apply data sheet templates to ensure that every supplier will be able to deliver the required information correctly and completely. Harmonize these templates with the customer and the contractor (and contractor suppliers).
− Take into account supporting analyses such as LORA for comparison of alternatives and/or SMA methods, such as S4000M, MSG-3, RCM, to build the maintenance concept for the IUA.
− Use simulation tools for those products performing missions according to operational profiles, focusing on maintenance and support while evaluating potential consequence from, for example, limited maintenance resources or changed operational profiles.
Depending on the different types of items, the following should be applied as a minimum analysis:

### Table 17 Depth of maintainability analysis depending on item type

<table>
<thead>
<tr>
<th>Item</th>
<th>Maintainability analysis depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items currently used under comparable conditions</td>
<td>Moderate information and data transformation, a more detailed maintainability analysis is voluntary</td>
</tr>
<tr>
<td>Items currently in use, but not under comparable conditions</td>
<td>Careful data transformation is mandatory, a more detailed maintainability analysis under new conditions is recommended</td>
</tr>
<tr>
<td>COTS items without major modification</td>
<td>Non-compliances of logistic features and/or requirements should to be documented and agreed by the customer.</td>
</tr>
<tr>
<td>Items currently available requiring minor modification</td>
<td>Moderate information and data transformation, a more detailed maintainability analysis is voluntary</td>
</tr>
<tr>
<td>Items currently available requiring major modification</td>
<td>Careful data transformation is mandatory, a more detailed maintainability analysis under new conditions is strongly recommended</td>
</tr>
<tr>
<td>Newly developed items based on well known technology</td>
<td>Detailed maintainability analysis is strongly recommended</td>
</tr>
<tr>
<td>Newly developed items based on new technology</td>
<td>Detailed maintainability analysis is mandatory</td>
</tr>
</tbody>
</table>

The results of the maintainability analysis must be carefully documented, therefore, maintainability reports should be defined and harmonized between the contractor and the customer. In addition to these reports, some results of the maintainability analysis will be documented within a logistic database within the maintenance concept reflected by the identified maintenance tasks.

### 7.2.7 Testability analysis

Main aspects for a testability analysis, which must be taken into consideration, are the following:

- Identification of maintainability strategy aspects, which must be observed
- Identification of overall test architecture and test principles
- Identification and verification of contractual testability requirements
- Evaluation of FMECA/FMEA documents concerning testability information
- Description of failure verification methods at equipment-, system- and end item level
- Description of functional check requirements at all involved levels
- Identification of related logistic resource requirements (personnel, loadable software, test software)

The testability analysis is closely connected to the maintainability analysis. Results from testability directly influence the maintainability analysis. Any corrective maintenance concerning test capable items is dependent of the detectability of the failure to be corrected. If a failure cannot be verified by any test procedure, the corresponding repair task cannot be performed. In addition to these aspects, it is necessary to be able to perform a functionality test of an item after repair. The depth of failure detectability/verification is one of the most important influencing factors on an applicable maintenance concept.

Since it is nearly impossible to establish testability features after the design of an item has been finalized, testability requirements must be described and documented carefully for each applicable item during the design and development phase. These requirements must be
harmonized with the customer during the LSA GC, however, testability capability should only be planned to the depth on which repair is intended.

For all items containing full Built in Test (BIT) capability, a testability analysis is considered mandatory. Related testability features must be identified and documented. A testability analysis report, as a summary of the analysis itself, is required. For equipment with reduced BIT capability items (eg off the shelf equipment) that are somehow monitored within the overall test architecture, a testability analysis is recommended. Related testability features must be identified and documented, also.

The types of data concerning testability features must be documented in a logistic database or in special testability reports and must be established and harmonized with the customer. Rules for manufacturers on how to implement testability features must be carefully established and verified (eg by testability demonstrations/validations together with the contactor and/or the customer).

7.2.8 Analysis activities concerning event driven maintenance

The following paragraphs describe the events that can be the justifications for maintenance activities. An overview of the event driven maintenance is given in Chap 12 in figure 1.

7.2.8.1 LSA FMEA (Logistic FMEA)
Each LSA candidate with an existing technical FMECA (eg MIL-STD-1629) must be considered as a potential full LSA candidate. The FMEA/FMECA results should be grouped as much as feasible in order to keep the required effort to a sensibly low level. This grouping will lead to a LSA FMEA or logistic FMEA. Criteria for grouping and the methods for documentation should be established and agreed to within the LSA GC. Grouping should be performed for every failure or defect that leads to the same maintenance action or chain of maintenance actions that have to be:

− performed on the same item
− at the same maintenance level
− by the same personnel (trade, skill level, number of persons)
− requiring the same material (special support equipment, spares)
− requiring the same user loadable software or data

7.2.8.2 Damage analysis
Items susceptible to damage should be considered as at least partial LSA candidates. In general, damages are special events, for which a prediction of criticality or extent cannot be given for every case. Nevertheless, it is useful to identify classes of damages, which can be maintained in the same way or which can be detected in the same way. Analyses that at least address initial diagnostic tasks and simple repair procedures (eg simple repair of damages such as scratches) are strongly recommended. The result of these analyses should be documented within general chapters in a logistic database, eg the standard test and repair procedures for structural items or for electrical wiring is collected within a non hardware chapter of the product breakdown.

Because reliability values concerning damages are not normally available, general quantitative statements are generally not possible. Only with the help of statistical evaluations is it possible to get a prediction of the required effort resulting from damages. It is recommended to use such predictions with caution.

7.2.8.3 Special event analysis
In additional to the occurrence of damages, which are a type of special event, other events can have an impact on the maintenance concept for an IUA or on the product. It is strongly recommended to identify and to document probable special events that require special maintenance tasks in order to guarantee the proper functionality for the further usage. Also, the
required maintenance tasks must be described and documented (including information concerning required resources like personnel, material or software) within a logistic database.

Some examples of special events that require maintenance tasks include:

− Exceeding temperature limitations
− Exceeding mechanical load limits (eg over torque)
− Exceeding maximum allowed speed
− Operation in salt-laden atmosphere
− Operation in sand-laden atmosphere
− Lightning strike
− Hard landing of an aircraft
− Collision with external objects (eg bird strike)

7.2.8.4 Scheduled maintenance analysis

A SMA is performed to ensure that safety relevant, highly economic or ecological failures are strictly avoided. This analysis is mandatory for safety relevant aspects and strongly recommended if significant economic or ecological disadvantages are to be expected. The SMA itself is an extensive analysis and the results have direct impact on the maintenance concept. Scheduled maintenance tasks, which are identified by the SMA from S4000M, MSG-3 or RCM processes, must be documented in a logistic database. It is recommended to establish rules on how to harmonize SMA results with the logistic database documentation method.

Each item for which scheduled or preventive maintenance tasks are identified via a SMA process should become a full LSA candidate.

For more detailed information on SMA, refer to Chap 10.

7.2.9 Level of repair analysis

Depending of the general support strategy, it may be necessary to decide for each item, the level at which maintenance and/or repair of the item will take place. To support this decision, a LORA can be performed by applying one of several methods, depending on the type of LSA candidate to be analyzed, the effort required and/or the information available. Depending on LSA GC agreements, different categories of LORA could be assigned, as shown in the examples in Table 18.

Table 18 Classification of different depths of LORA

<table>
<thead>
<tr>
<th>LORA classification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplified LORA</td>
<td>May be derived on the basis of Best Engineering Judgment by simply taking into account best information available. The results of this analysis must be documented and harmonized with the customer through, for example, a &quot;Simplified LORA report&quot;.</td>
</tr>
<tr>
<td>Full LORA</td>
<td>A front end analysis using software packages available based on mathematical models of different complexity and accuracy. All required data concerning the item itself and the context of its use (eg operational scenario, spare part provision, costs) must exist in order to perform such an intensive analysis.</td>
</tr>
<tr>
<td>Analysis via simulation</td>
<td>A simulation that considers very detailed information about the IUA and its deployment and operational scenario may produce the best LORA results. This requires a complete set of information/data in the required format for the simulation software package to be used. The preparation of this data can cause a lot of additional effort.</td>
</tr>
</tbody>
</table>
In any case, where applicable, a LORA analysis must be considered to be a mandatory analysis activity. This holds for LSA candidates with alternative repair and/or support solutions concerning personnel, material and/or user loadable software or data as well as for scheduled maintenance tasks derived from an SMA or equivalent and/or servicing tasks, if contracted.

7.2.10 Maintenance task analysis
The MTA is one of the central analysis activities within the LSA process. Here, the identified maintenance tasks (both scheduled and unscheduled) are detailed with all required information available. The following list gives a limited overview, for more detailed information refer to Chap12.

- Documentation of general task information such as preconditions for task performance, training requirements or criticality information.
- Assignment of maintenance tasks to the identified events (eg failures, damages, special events, time limitations)
- Rough task description (sequence of subtask)
- Identification of related logistic resource requirements (eg personnel, support equipment, spares, facilities, software)
- Time estimations
- Calculation of task frequencies
- Consideration of required pre- and post- tasks (eg test, fault location, gaining access)

7.2.11 Software support analysis
Each item containing user loadable data/software must be analyzed with regards to loading, de-loading, transportation and documentation of existing SW releases. This covers maintenance and servicing tasks applicable for:

- Maintenance of hardware containing user loadable data/software
- Data and/or operational software required for usage preparation
- Software support in order to update items with new user loadable software releases

If a SSA program will manage SW changes, SW releases, and the maintenance of SW in case of SW failures (bug fixing), a process analogue to the LSA process for hardware should be established.

For more detailed information, refer to Chap 13.

7.2.12 Operations analysis
Operational requirements must be analyzed to identify tasks that are important for the general handling of the SUA. These tasks should also be documented in a logistic database. For logistic purposes, it is necessary to analyze such tasks to identify the requirements concerning personnel, support equipment or material.

For more detailed information, refer to Chap 9.

7.2.13 Simulation of operational scenarios
This analysis will be performed by special request only. Simulation represents a very powerful ability to evaluate operational scenarios in combination with logistic scenarios. Today, simulation SW packages offer the opportunity to analyze and optimize the planned usage of a system under particular logistic conditions. Only the combination of these two aspects, planned usage scenario and planned logistic scenario, makes it possible to generate meaningful analysis results. The better the scenario can be described, the better the results will be.

It must be clear that simulation analysis needs a lot of input data with a certain depth. This causes additional effort, which must be considered. If simulation is selected as a required analysis activity, it is strongly recommended to start early with the examination of the data needs of the simulation SW package. The decision to perform simulation can influence the
depth and the requirements for the quality of any logistic data (e.g., the detailed description of a logistic scenario can cause additional effort within a MTA). Also, the creation of an operational scenario can cause additional effort for the customer.

7.2.14 Training needs analysis

Within this analysis, it must be decided whether a task requires special training or not. If training is required, it must be determined how the training can be applied most effectively. This process can be supported with the help of the content of the LSA database concerning the identified tasks.

The criteria for a training requirement should be discussed and harmonized between the customer and the contractor during the LSA GC. It is recommended to establish an IT supported process for the TNA. The criteria should be "translated" to the LSA database system and each task should be examined against these criteria. The result can be a preliminary TNA decision based on existing data concerning tasks within a LSA database.

7.3 Analysis relations and general overview

All the analysis types described in the previous paragraph are interconnected and influence each other. Some analysis types are a basic precondition for other ongoing analysis processes. Some analysis types are of a general significance for all other analysis procedures. Fig 19 shows a graphical depiction of the relations between the different logistic analyses.
7.4 Analysis activities selection criteria

The following aspects must be considered in order to select and allocate LSA relevant analysis activities to LSA candidates that are applicable and effective depending on the type of items, candidates and required information.

The selection criteria and aspects must be tailored to each program by taking into account any specific circumstances and they must be harmonized between the contractor/industry partner companies and the customer within the LSA GC.
### Table 19 Analysis activities selection criteria

<table>
<thead>
<tr>
<th>Criteria group</th>
<th>Criteria</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items with existing experience</td>
<td>Items already in use under comparable conditions</td>
<td>Moderate transformation of existing analysis data</td>
</tr>
<tr>
<td></td>
<td>Items already in use but under non-comparable conditions</td>
<td>Careful transformation of existing analysis data</td>
</tr>
<tr>
<td></td>
<td>COTS items usable without major modification</td>
<td>Moderate transformation of existing analysis data</td>
</tr>
<tr>
<td></td>
<td>Items currently available that would require (minor/major) modification</td>
<td>Careful transformation of existing analysis data or performance of new analysis</td>
</tr>
<tr>
<td></td>
<td>Newly developed items based on well known technology</td>
<td>Performance of analysis activities in required depth strongly recommended</td>
</tr>
<tr>
<td></td>
<td>Newly developed items based on new technology</td>
<td>Performance of analysis activities in required depth mandatory</td>
</tr>
<tr>
<td>Items that need to be assessed for global point of view or for special interests</td>
<td>Potential readiness drivers and/or cost drivers</td>
<td>Criteria for these items must be detailed within the LSA GC</td>
</tr>
<tr>
<td></td>
<td>Subject to emphatic customer interest</td>
<td>Items must be detailed within the LSA GC</td>
</tr>
<tr>
<td></td>
<td>Subject to LSA related contractual fulfillment</td>
<td>Items must be detailed within the LSA GC</td>
</tr>
<tr>
<td>Items that need to be assessed for inherent reasons or due to the installation area</td>
<td>Item is maintenance significant concerning the related failure/defect frequency, workload, special logistic requirements (personnel and/or material)</td>
<td>Criteria for these items must be detailed within the LSA GC</td>
</tr>
<tr>
<td></td>
<td>Item is subject to scheduled maintenance or to life limits</td>
<td>Criteria for SMA must be detailed within the LSA GC</td>
</tr>
<tr>
<td></td>
<td>Item is subject to preventive maintenance resulting from special events</td>
<td>Criteria for special event analysis must be detailed within the LSA GC</td>
</tr>
<tr>
<td></td>
<td>Item is potentially susceptible to damages due to its installation area</td>
<td>Criteria for damage analysis must be detailed within the LSA GC</td>
</tr>
<tr>
<td>Type of LSA candidates</td>
<td>Full candidate</td>
<td>Subject to applicable analysis, results documented within LSA</td>
</tr>
<tr>
<td></td>
<td>Partial candidate</td>
<td>Rudimentary information to be documented in LSA</td>
</tr>
<tr>
<td></td>
<td>Candidate families</td>
<td>A group of items considered as one LSA candidate, subject to applicable analysis, results documented within LSA</td>
</tr>
</tbody>
</table>

**Applicable to:** All
<table>
<thead>
<tr>
<th>Criteria group</th>
<th>Criteria</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items subject to standard procedures</td>
<td>Rudimentary information to be documented in LSA</td>
<td></td>
</tr>
<tr>
<td>Information expected from the LSA process</td>
<td>To be harmonized with and agreed to by the customer during the LSA GC</td>
<td></td>
</tr>
<tr>
<td>Proposal of recommended logistic support principles</td>
<td>To be harmonized with and agreed to by the customer during the LSA GC</td>
<td></td>
</tr>
<tr>
<td>Identification of possible alternatives (for hardware, software, support concepts) and related consequences (eg MC, logistic support requirements)</td>
<td>To be harmonized with and agreed to by the customer during the LSA GC</td>
<td></td>
</tr>
<tr>
<td>Proposal of recommended maintenance concepts including corresponding tasks</td>
<td>To be harmonized with and agreed to by the customer during the LSA GC</td>
<td></td>
</tr>
<tr>
<td>Estimation of related logistic support costs</td>
<td>Identification of related logistic support requirements</td>
<td>Estimation of related logistic support costs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Information for early system/program decisions (significant costs influence)</td>
</tr>
</tbody>
</table>

### 7.5 Checklist for analysis activity recommendation

#### 7.5.1 Recommendation and decision sheet

For analysis activities selection purposes, a selection matrix should be used in order to structure and document the selection and decision results.

Analysis activities recommendation sheet:

This matrix should be prepared by the contractor for the LSA GC, identifying the LSA candidates, the task selection criteria and the derived results (recommendations). For all general tasks not driven by the LSA candidates, recommendations must be documented carefully eg in independent documents. This applies to the following analysis activities:

- Analysis for identification of general LSA needs
- Configuration assessment/aspects
- Availability of a technical FMEA/FMECA and influence on LSA FMEA and SMA
- Human factor analysis

Analysis activities decision sheet:

This matrix will be harmonized and finalized at the LSA GC and reflects the decisions made by the customer. This contractually relevant document should be established as a living document in order to reflect changes, to incorporate "lessons learned" results and enable traceability during the life cycles of the program. As a recommendation, a preliminary list could become part of a commercial proposal in order to introduce a general guideline for LSA task allocation to the customer for risk reduction purposes.

#### 7.5.2 Selection matrix example and rating values

In order to identify the relative importance of the allocated analysis activities, "ratings" should be agreed to concerning both the importance of the selected LSA candidate itself as well as the importance of the potential analysis activity. The result can rank LSA candidates and analysis
activities from mandatory down to voluntary. This ranking could be used to refine the selection of candidates/analysis activities in case of e.g. limitations of budget or time constraints. The results of this selection procedure would be the initial issue of the LSA CIL as well as the range of LSA activities considered as relevant and effective for the LSA candidates. Appropriate rules must be identified and harmonized within the industry partner companies and agreed to by the customer during the LSA GC.

**Fig 20 Example of a simple analysis activity recommendation sheet**

*Fig 20* shows a simplified example of a recommendation matrix for the selection of analysis activities. For an actual project, it may be necessary to go into further detail, for example, the performance of a LORA can be divided up in more categories such as simplified LORA or full LORA with the help of a special software package. Additionally, the rating values can be more detailed.

The creation of the rating method should only be done by very experienced personnel in order to ensure that the result of the recommendation sheet for analysis activities selection is reliable and sensible.

### 7.6 Analysis workflow processes

In the different project phases, different sets of information are available for the execution of the logistic analysis activities. Some of the tasks described in the paragraph above can typically be addressed during an early phase of the project. Some of the tasks can be executed at a stage in which detailed and final information concerning the design are available (e.g. an extensive...
SMA can be done during a late stage of design, because a lot of detailed information is required for this type of analysis.

### Early phases

<table>
<thead>
<tr>
<th>Conception phase</th>
<th>Risk reduction phase</th>
<th>Design &amp; development phase</th>
<th>Production and introduction phase</th>
<th>In-service phase</th>
<th>Waste disposal phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>First logistic analysis tasks</td>
<td>Analysis for identification of general LSA needs</td>
<td>Comparative analysis</td>
<td>Human factor analysis</td>
<td>Operations analysis</td>
<td>Configuration assessment</td>
</tr>
</tbody>
</table>

### Detailed logistic analysis tasks
- Configuration assessment
- Reliability analysis
- Maintainability analysis
- Testability analysis
- LSA FMEA (Logistic FMEA)
- Damage analysis
- Special event analysis
- Scheduled Maintenance Analysis (SMA)
- Level of Repair Analysis (LORA)
- Maintenance Task Analysis (MTA)
- Software Support Analysis (SSA)
- Operations analysis
- Training Needs Analysis (TNA)

**Fig 21 Position of LSA in the overall project schedule**

Logistic analysis requirements do not end at the end of a design & development phase. Within the next phase (production and introduction), logistic analysis can be required multiple times, depending on the modifications being made to the product.

This is especially relevant for the in-service phase, which is normally the longest phase. There will be extensive modifications during the in-service phase and, depending on the modification, logistic analysis will be necessary again in corresponding depth. Therefore, the iterative LSA/ILS process will reappear repeatedly. For this reason, it is recommended that the documentation of logistic data and logistic decisions is continued throughout the entire project, ending with the disposal phase. Even for this final phase, logistic aspects must be considered.

The next paragraphs give a general impression on how the different analysis activities can be typically assigned to the project phases.

#### 7.6.1 Early phase analysis activities

In the early phase of a project, the logistic analysis activities can be limited to the ones that only require a rather low level of information. Before the LSA GC, information about the operational and customer requirements is normally available and drafts for basic rules of the LSA process are prepared.

Detailed information of the SUA or the IUA is not normally available. Only basic conditions and rules for the application of the process should be available in the early stages, nevertheless, some analysis activities can, or even should, start in the early stages of the LSA process:

- Analysis for identification of general LSA needs
- Comparative analysis
- Human factor analysis
- Operations analysis
- Configuration assessment

7.6.2 Design freeze analysis activities
During the Design & Development phase, the LSA process should accompany the design process. This fits for almost all LSA activities with the exception of those activities that should be performed in a very early phase or after a design freeze. The following activities should accompany the design & development phase:

- Configuration assessment
- Reliability analysis assessment
- Maintainability analysis
- Testability analysis
- LSA FMEA (logistic FMEA)
- Damage analysis
- Special event analysis
- SMA - Scheduled Maintenance Analysis (S4000M, MSG-3, RCM)
- LORA - Level of Repair Analysis
- MTA - Maintenance Task Analysis
- Software and data uploading/downloading/transportation analysis
- SSA - Software Support Analysis (software design and software maintenance)
- Operations analysis

After design freeze, the collection of logistic relevant data should be mostly complete. For this reason, analysis activities that need a lot of detailed data should be performed at a late stage of the design process in order to guarantee a fixed design and to reduce the risk of repeating costly analysis activities. This applies for:

- Simulation operational scenarios
- TNA

7.6.3 Inservice phase analysis activities
During the inservice phase, the logistic analysis activities will repeat, but to a lesser extent. If the product has technical upgrades or design changes, a new analysis concerning logistic requirements should be performed. The depth and extent of this analysis will depend on the type of system and the scale of change.

8 Customer involvement
In order to avoid negative consequences of misunderstandings between the contractor and the customer, the customer shall be involved in the industry interpretations and approaches in an appropriate depth during the whole LSA program. The customer involvement with regards to the selection of LSA candidates, relevant analysis activities and resulting analysis data shall be covered as well as the principles of information exchange, documentation of results and related management aspects.

8.1 Customer assessment of candidate items and recommended analysis activities
The selection of LSA candidates in conjunction with the allocation of appropriate analysis activities must be considered as the most cost influencing for a LSA process. For that reason, these selections must be done and assessed with great care in an early program stage. In order to reduce risk, an initial selection should be done prior to the signing of the contract.

In general, the contractor must select the LSA candidates and related LSA relevant analysis activities based on a product breakdown as detailed in Para 5 for establishing of breakdown,
Para 6 for LSA candidate selection, and Para 7 for recommended analysis activities. The result must be released to the customer for assessment.

Note
Since a list that only reflects the proposed LSA candidate items (CIL) would not provide a clear picture of what it is really intended, it is strongly recommended to assess both the proposed Candidate Items (CI) and the related analysis activities in order to find a well balanced decision.

8.1.1 Establishing LSA assessment rules
The assessment of proposals must be done under the customer's responsibility, and some general rules concerning principles for assessment should be established, as a minimum, but not limited to:

- Only aspects that concern rules agreed to in the LSA GC or rules that may be agreed to between the customer and the contractor within the subsequent LSA process (eg in LSA reviews), must be considered as assessment relevant.
- If an assessment has been performed and has reached a successful final clarification, it must not be reassessed unless new assessment relevant aspects occur.
- Only rules that have been established within the area of the LSA community must be considered as assessment relevant.
- Other communities (eg technical documentation, materiel support, and training) are not allowed to establish or define subjects with relevance for the LSA assessment.
- A change in the customer's staff is not to be considered as assessment relevant and must not be accepted by the contractor as a reason for any reassessment.
- Rules concerning time limits between the release of LSA deliveries and an adequate response should be established and agreed upon, along with the consequences of non-fulfillment.
- Both the invitation for assessment of details within the contractor's LSA deliverables as well as the assessment result should be indicated using applicable codes within a "status code".
- The structure, details of information and individual codes to be used are the responsibility of the contractor but must be coordinated with the customer.
- The LSA review structure should be organized in line with the information group represented by the status code.
- An appropriate data element within the logistic database should be established for "status code".
- The status code concerning each LSA candidate must be kept updated as applicable in the logistic database.

8.1.2 Initial assessment and re-assessments
8.1.2.1 Initial assessment
During the initial assessment, primarily the principles have to be considered such as:

- Is the product breakdown considered to be in line with the rules established during the LSA GC, eg especially concerning its structure, content and depth?
- Are rules for selection of LSA candidates and for non selection or non recommendation available and sufficient?
- Is the selection of LSA candidates and relevant analysis activities selection in line with rules established within the LSA GC?
- Assessment results must be documented and passed to the contractor according to the established assessment procedure and followed until a final clarification status is reached.
- Once assessed and commented upon and final clarification is reached between the customer and the contractor, those principles must not re-assessed or re-discussed for later assessments.
8.1.2.2 Re-assessments
Assessments subsequent to the initial assessment must concentrate on aspects such as:

− Items released by the contractor designated for assessment
− Assessment of contractor deliveries concerning general questions or comments (new aspects)
− Assessment of contractor deliveries concerning detailed questions or comments (observations)
− Assessment of contractor deliveries concerning details such as like discrepancies with prior agreements
− Assessment results must be documented and passed to the contractor according to the established detailed assessment procedure and followed until a final clarification status is reached.

8.1.3 Establishing an assessment procedure
A detailed assessment procedure (including an appropriate commenting process) shall be established that is tailored to the specific requirements of the individual LSA program and comprises the above elements. This procedure should be harmonized between the customer and contractor.

8.2 Information exchange between customer and contractor
The customer should inform the contractor about the individual assessment results that are relevant in order to proceed with the LSA process, mainly on agreements/disagreements together with the reasons for any disagreement. In addition, general comments covering global aspects and/or general questions may occur. Those comments and questions should be kept separate from the other comments as they usually need a special response such as a dedicated explanation or request for special training.

The content of this paragraph is not limited to LSA candidates and related analysis activities, but it also applies to detailed results of the analysis and to a response on any official LSA report.

8.2.1 Commenting process
A structured commenting process should be established in order to ease the management of comments and to enable traceability.

− Any LSA delivery from the contractor to the customer released for assessment shall be registered and officially commented upon.
− Any customer agreement must be documented carefully.
− Any customer disagreement must be documented along with the related reasons. Affected items must be monitored until final clarification has been reached and the clarification is documented.

In order to cover these subjects, the commenting process shall be detailed as follows.

8.2.1.1 Registration of LSA deliverables
Any LSA delivery that needs assessment shall be registered according to rules and files that have been agreed to and established as part of the LSA/ILS management structure.

8.2.1.2 Registration of comments
For each LSA delivery released for assessment, the customer response must be documented. For that reason, a status must be registered for each applicable BE depending on the consequences of the customer response.

8.2.1.3 Registration of the LSA status
The LSA status code should be structured in order to belong to different information requirements per released BE (refer to Para 9.4)
8.2.2 Informing the contractor by the customer

The customer informs the contractor as agreed in the established commenting process about:

- Agreements concerning released LSA deliverables: This will be documented by the contractor according to established management rules. In the case of a consortium, all companies involved in the contractor network must be informed according to related rules.

- Disagreements concerning released LSA deliverables: In these cases, the response will be investigated by the contractor (or the involved company) in order to identify an adequate response (e.g., further explanation, proposed rework by industry). The contractor’s response will be forwarded to the customer in order to reach an agreement in an iterative manner.

- When issues remain open and cannot be resolved by iterative commenting, an adequate solution must be identified in order to reach a final clarification.

---

**Table 20 Explanation of time schedule of the commenting process**

<table>
<thead>
<tr>
<th>Time</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₀</td>
<td>The start of the preparation of the LSA delivery items by the contractor. This preparation can include the integration and harmonization of data between more than one contractor in case of a contractor consortium.</td>
</tr>
<tr>
<td>T₁</td>
<td>Delivery of the contractual LSA delivery items to the customer in the agreed format.</td>
</tr>
<tr>
<td>T₂</td>
<td>Between T₂ and T₁, the commenting of the customer on the delivered items (e.g., LSA data, LSA reports) will take place. At the agreed point of time T₂, the delivery of the comments to the contractor will take place.</td>
</tr>
</tbody>
</table>
To reduce the effort of the LSA review, any type of comments that can be answered easily by the contractor should be handled before the LSA review, if possible. Nevertheless, any decision concerning the customer's comments must be documented carefully.

During this time, more than one question-answer loop can clarify special questions in detail, however, the number of loops should be limited since more extensive questions must be clarified (e.g., at the LSA review or by a longer clarification process between the contractor and customer).

Taking into account the clarification loops between the customer and the contractor in the time between T₂ and T₄, the remaining customer comments will be delivered to the contractor. These comments will be the basis for any discussions and clarifications in the LSA review.

Time of LSA review. It must be ensured that every decision in the LSA review is documented in an update of corresponding status information.

8.3 Final clarification on open issues
For any remaining open issues, an adequate solution must be identified. If this is not possible during a regular review session, a meeting of dedicated specialists could be required to reach a satisfactory solution.

8.4 Customer decision (status influencing)
Normally, it is required to distribute a LSA delivery for assessment to different places (e.g., different authorities in different nations and, since LSA is to be understood as inter-disciplinary, to all involved logistics communities). Subsequently, individual comments must be collected and (in case of deviating responses) be harmonized in order to provide one official customer response.

Note
It is important for the contractor to receive only one official customer response in order to proceed with the LSA process in a manageable way. It is recommended to receive the customer's decisions as a status code change (following the rules established for status code allocation).

The status code allocated by the customer remains unchanged in the related customer release. Any related explanation (e.g., reason for disagreement) also must be documented as part of the contractor response and remain untouched for traceability reasons.

For LSA program traceability reasons, both, the LSA delivered as proposed for assessment, as well as the customer's response (if applicable including final clarification results), may be documented as:

- LSA recommendation document delivered by the contractor, and
- LSA decision document responded to by the customer

Especially in the case of essential documents such as LORA or SMA results, the resulting recommendation/decision documents may ease some clarification requirements.

Within the contractor's LSA database, corresponding status codes concerning the involved breakdown elements must be updated accordingly. However, contractor's information may supersede the status code as allocated by the customer (e.g., on-going work status, required deletions).
8.5 Exchange of analysis data with the customer
In general, the following steps must be applied:

8.5.1 Establishment of appropriate rules for data and document exchange with the customer
Appropriate agreements and rules must be documented and agreed to, by the customer and the contractor in the LSA GC and the subsequent LSA agreements, for example:

− Software to be used
− Data and report formats (eg DEX-format)
− Periodicity
− Other reasons for delivery
− Distribution partners and sequences to be obliged

8.5.2 LSA data and document collection by the contractor
− Establishment of an appropriate data/document exchange and file network within industry
− Collection of data/documents for intended delivery between associated companies and LSA related disciplines
− Performance of industry internal quality checks, harmonization and agreement for delivery

8.5.3 Delivery of LSA data/documents by the contractor
− Distribution of the LSA deliverable (directly or via a designated management unit) by taking into account the agreed distribution rules (as well as the due date, if applicable)
− Contractor internal documentation and distribution of the official LSA delivery.

8.5.4 Customer assessment and distribution of assessment results
− Distribution of the contractor's LSA delivery, as required
− Harmonization of the individual responses to the official customer response
− Distribution of the customer response to the contractor

8.5.5 Distribution of customer response by the contractor
− Registration of the official customer response
− Industry internal distribution
− Identify, distribute and harmonize the contractor investigation results
− Update of the LSA database according to the customer response details (as much as possible)
− Preparation of harmonized contractor response to general comments and disagreements

8.5.6 Distribution of the contractor response concerning disagreed comments
Steps concerning iterated analysis data from the contractor as described in Para 8.5.3.

9 LSA review conference
A LSA Review Conference (LSA RC) should be planned with the participation of contractor and customer personnel in order to:

− reach final clarification of open issues
− reach customer acceptance on open issues
− monitor the status of LSA candidate items, eg concerning completion and quality
− assess logistic support aspects related to each phase of the process
− reach additional agreements between the customer and the contractor concerning LSA aspects in order to improve the LSA process, as required

9.1 LSA review process overview
The LSA RC shall be conducted at LSA candidate level. Prior to the conference, the contractor shall inform the customer about the LSA candidates to be discussed.
During the LSA RC, the related LSA deliveries and the relevant data elements shall be assessed and decided upon by taking into account established acceptance criteria. Any decisions must be documented.

Complex LSA candidate items may require a sequence of analysis activities, which may be distributed over a long period of time, depending on the type of information that is available within different program stages. On the other hand, some decisions must be made in a very early program stage based on limited information (e.g., establishment of a preliminary maintenance concept) while other decisions require a stable design that is not typically available until a very late stage (e.g., identification of scheduled maintenance tasks based on time-consuming, excessive analysis like MSG-3). For this reason, different review steps must be established in order to enable sequenced decisions in line with the structured analysis process. Therefore, the overall review process must be divided into different review steps, each comprising coherent information. These review steps shall be identified by review step indicators (refer to Para 9.3). It should be defined which data elements must be available for or must be assessed during the individual review steps. Each LSA RC should cover any steps that belong to the individual LSA candidates being requested for assessment which will be indicated by according status codes.

Due to the iterative nature of any LSA process, any approval given during a review conference must be considered as a temporary assessment but it must allow the contractor to continue with the LSA process. As a consequence of the iterative nature of LSA, the data may be re-examined due to changes in the design, the updating of technical data and/or the support or operational environment.

The final results to be documented in the LSA documents and/or within the LSA database, e.g., at the end of each program phase, must be considered as frozen. Related acceptance criteria must be agreed to in the LSA GC.

The LSA RC should be held in accordance with the evolving design and the progress of the intended analysis activities and they should take place regularly within an agreed period in time or depending on the amount of information to be assessed. Minutes of each LSA RC must be prepared to show the results and to give evidence of the necessary actions for the customer and the contractor. The updating of related LSA documents and/or the LSA database in accordance with the results of a LSA RC must be considered a permanent action.

9.2 Subject to be reviewed

In general, any contractual item agreed to during a LSA GC, or by subsequent agreements between the customer and the contractor must be reviewed within a LSA RC. In addition, any aspect that can be identified as essential for the LSA process or that may influence other logistic processes with interfaces to the established LSA program may be discussed within a LSA RC in order to agree on modified or additional LSA tasks. Highly unlikely occurrences within the LSA process that require common assessment in order to correct any essential problem should be coordinated between the customer and the contractor.

Introductions into procedures and/or principles of LSA relevant analysis required should be presented by the contractor to the customer in order to reach a common understanding concerning the goal of analysis activities, related restrictions and essential LSA results. This includes the introduction into global analysis work flowcharts.

Progress reports and data quality assessment summaries should also be provided by the contractor, as well as summary reports reflecting items of special interest for the customer. Some examples include:

- Maintainability values such as mean man-hours per operating hour
- Logistic performance parameter such as Mean Elapsed Time (MET) for maintenance tasks that occurred within specified limits, along with the related percentage
9.3 Examples for review structuring

The following examples provide an idea of how a typical review process may be structured. Different aspects and/or types of information at different stages of the overall analysis process will be covered in this way. The following steps can be arranged as required by a project to realize LSA data release and acceptance within the LSA review process.

9.3.1 LSA review step - CIL and maintenance analysis allocation

This step forms the basis for all subsequent LSA tasks documented within the LSA database and consists of:

- BEI and LSA candidate selection (including grouping of candidates, where applicable)
- Identification of the type of candidate and related attributes, such as:
  - Identification of LRUs
  - Potential cost drivers
  - Potential maintenance drivers
  - Potential readiness drivers
- Allocation of LSA relevant analysis activities to be performed for each selected candidate
- Status code reflecting details
- Any updates regarding changes in the design configuration at the candidate item level must be managed in the LSA database

9.3.2 LSA review step - LORA results and maintenance concept information

In this step the (preliminary) maintenance concept will be prepared by allocating maintenance tasks to their related maintenance levels and performance sites. This task typically includes:

- Maintenance concept recommendation proposed by the contractor.
- Identification of maintenance tasks relevant for equipment integration into the product, such as:
  - Gain and undo access
  - Remove and install equipment
  - Performance of functional tests at equipment and system level, as required
- Substantiation concerning the proposed maintenance concept as required (eg LORA results, applied method or procedure, main influencing aspects to be considered, other analysis being applied or assessed for LORA constitution).
- Customer decision on the maintenance concept (rejection or alternative solution)

9.3.3 LSA review step - LSA FMEA and SMA results

In this step, the maintenance concept shall be finalized by taking into account the LSA FMEA (eg constituted from technical FMECA by the grouping of failures focused on the intended maintenance tasks) for the identification of unscheduled maintenance tasks. Results from SMA will identify scheduled maintenance tasks.

9.3.4 LSA review step - Maintenance Task Analysis (MTA)

In this step, the identified maintenance tasks will be analyzed concerning its requirements and its execution:

- Description of the required maintenance tasks in applicable detail and depth
- Duration of each task step
- Identification of related logistic resources required to support the maintenance activities (eg personnel, support equipment, spare parts, consumables, facilities, data and software)
9.4 Status code examples

In order to reflect the status of each LSA candidate in detail concerning the identified review steps, a status code should be used, which may consist of sub codes reflecting the following information:

- Responsibility of involved companies. For each task group, the responsible company should be identified. This also reflects the agreed work share details concerning this subject.
- Type of LSA candidate (eg full, partial)
- Important aspects (eg contains user loadable SW and/or data)
- Review step indicator
  Here, the analysis status of the addressed LSA candidates must be indicated concerning the content of each review step (eg as described in Para 9.3).

The following table gives examples for potential codes and the corresponding meaning (for each LSA candidate review step indicator):

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Review step content is in &quot;working condition&quot;, assessment not requested</td>
</tr>
<tr>
<td>N</td>
<td>Review step is &quot;not applicable&quot; for the related LSA candidate</td>
</tr>
<tr>
<td>R</td>
<td>Review step content is requested for assessment and decision in the next LSA RC</td>
</tr>
<tr>
<td>A</td>
<td>Review step content is (temporarily) agreed to by the customer</td>
</tr>
<tr>
<td>B</td>
<td>Review step content was assessed and in principle (temporarily) agreed to by the customer. However, minor rework is expected prior to final acceptance. In terms of contractual status, a &quot;B&quot; can be treated like an &quot;A&quot;.</td>
</tr>
<tr>
<td>O</td>
<td>Review step content was not agreed to by the customer and remains an open issue</td>
</tr>
</tbody>
</table>

9.5 Depth of status code allocation

As a minimum requirement for each LSA candidate, relevant status codes should be allocated. In addition, the subject of assessment may be addressed in more detail (eg the specific task of the LSA candidate that was established/modified and needs to be assessed). This decision must be made during the LSA GC.

9.6 Functions of the status code

The status code shall reflect the review process structure and address all contractual issues that have to be documented within the LSA database. It can be used for overall status summary reports concerning the content of the LSA database, eg after each LSA RC to reflect the progress in the area of LSA.

A LSA candidate list filtered by status code may be used to focus the customer's attention to this specific subject.

The status code may also be used internally by the contractor in order to indicate to logistic disciplines the "stop" or "go", respectively to indicate any risk when creating ILS products.

9.7 Status code definition at the LSA guidance conference

The intended review steps, as well as the related status code structure and applicable status codes, must be proposed by the contractor and harmonized with the customer within the LSA GC along with the depth (level of detail) of status code allocation.
10 Starting point and management of the creation of the logistic products

10.1 Starting point recommendations
The start point for the creation of the logistic products is dependent on several factors, including technical documentation and illustrated parts data availability. Changes of design or maintenance concepts can have also extensive influence on these products. In this context, ILS products to be considered include:

- Technical documentation
- Materiel support (illustrated parts data)
- General and special support equipment
- Training
- The facility construction is not considered a typical ILS product since the decisions for facilities are made in an early phase of a project, because of the long qualifying time.

10.1.1 Technical documentation
In general, technical documentation consists of two main parts:

- Handbooks for system repair and maintenance
- Handbooks for system operation (user documentation)

During a design and development process, the supporting logistic analysis activities are performed. Depending on the progress of these analysis activities, it becomes clearer which maintenance concept will be the basis, which personnel, support equipment and spares are required and how the task will be executed (task description). For this reason, technical documentation concerning system repair and maintenance should start at a later point. The better the information from the logistic analysis activities is, the less the risk for the creation of the technical documentation remains.

Nevertheless, experience has shown, that creation of technical documentation must start early because of time limitations and contractual preconditions. The basis for the creation of technical documentation concerning system repair and maintenance is the maintenance task, which is documented in the LSA database. It is recommended to find a proper solution to trigger the technical documentation to begin the creation of real products. This process must be harmonized between the contractor’s support engineering and the technical documentation departments. A good solution is to use status information for LSA candidates, or even for maintenance tasks, within the LSA database.

The general, the correlation between LSA and technical documentation can be seen in Fig 23.
Generally, the creation of the handbooks for product operation is independent from the LSA process. It must be guaranteed by the technical documentation department that this part of documentation is available early enough to operate, for example, a prototype of a technical system safely for testing and qualification needs. A special aspect is the consideration of information from the operational analysis activities. The tasks described and documented in the LSA database in this area can be part of both the handbooks for product repair and maintenance, and the handbooks for product operation. In this case, the recommendations for the repair and maintenance documentation are also valid for the results of operational analysis.

It must be clear that each repair or maintenance task identified and documented in the LSA database must have its corresponding part of technical documentation in the handbooks for product repair and maintenance.

Nevertheless, the handbooks for system repair and maintenance can contain information beyond the documented LSA tasks, because in general it is not possible to include each single maintenance action within the LSA database (e.g. because of budgetary reasons). Not every piece part will be a LSA candidate. The purpose of LSA should be to analyze, in depth, the true maintenance and cost drivers. The remaining equipment or piece parts that require technical documentation must also be considered. The depth and the extent of this additional technical documentation should be clarified between the contractor and the customer within the Guidance Conference technical documentation.

To guarantee a meaningful information flow between LSA and technical documentation, it is recommended to establish a process to forward the status of work of LSA to the technical documentation department on a regular basis. In this way, technical documentation can find all relevant information needed to start production. The following aspects should be covered:

- Is the relevant maintenance task ready for technical documentation?
- If the status of the LSA task indicates: "Not ready for technical documentation", what risk could be taken to start the production of technical documentation anyway?
− Is there a blocking status in LSA, which permits the start the production of technical documentation?
− Which additional activities must be documented within technical documentation, which are not a part of the LSA database (no trigger from LSA for those aspects)?

10.1.2 Materiel support

The identification of relevant spare parts is a main task within the LSA process. The depth of the breakdown and the extent of analysis activities will be reflected in the identification of spare parts. Depending on the maintenance concept, identification of spare parts can differ drastically. For example, in a simple 2-level maintenance scenario, complete components will be replaced and no repair activities will be performed by the operator of the system. In this case, only a small number of spare parts will be identified. However, additional spare parts such as standard parts, could be identified that are additionally required by the identified complete components.

The decisive question is the depth of the breakdown within the LSA. Theoretically, a breakdown down to the last piece part is possible, but the effort would be irresponsible. Therefore, in practice, the identification of spare parts within the LSA process will stop at a certain level. Typically, the maintenance drivers and the cost drivers should be identified within the LSA process. This information is documented within the requirements of maintenance tasks in the LSA database. A similar process as described for technical documentation should also be established for materiel support. Depending on the progress of the logistic analysis activities, the start of the creation of the logistic products concerning materiel support should be triggered by LSA relevant maintenance drivers and cost drivers.

Similar to technical documentation, it should be clear that not all required spare parts will be identified by LSA. However, all spare parts identified by a LSA task must be considered within materiel support. The following table gives an overview of the different types of spares and their consideration in the identification process.

<table>
<thead>
<tr>
<th>Spare part type</th>
<th>Description</th>
<th>Correlation between LSA and materiel support</th>
</tr>
</thead>
</table>
| Complete piece of equipment | - Maintenance driver  
|                    | - Cost driver  
| Required spare parts for replace tasks (scheduled or unscheduled) at operator site (replacement of complete equipment without any repair at operator site) | Identification of the equipment as a required spare part should be approved by a maintenance task defined within the LSA. |
| Dedicated spare parts | Required spare parts for equipment repair or maintenance tasks at operator site (repair of the equipment will be performed at operator site) | Identification of this component as a required spare part should be approved by a maintenance task defined within the LSA. |
| Standard parts | Required standard parts (eg attaching parts or seals) for repair or maintenance tasks at operator site (repair of the equipment will be performed at operator site) | Identification of this component as a required spare part should be approved by materiel support because of missing depth of breakdown within the LSA. |
| Consumables | Required consumables such as liquids, grease, special chemical products, source material, glue. | Identification of this component as a required spare part can be approved by materiel support or by LSA. |
The process to begin the preparation of illustrated parts data within materiel support is shown in Fig 24:

**Fig 24  Correlation between LSA and materiel support**

### 10.1.3 Common and special support/test equipment

The identification of relevant support and test equipment is a main task within the LSA process. The timely provisioning with required support/test equipment is of crucial importance and it should be distinguished between common and special support/test equipment. Especially requirements for special support/test equipment should be identified by the LSA process. After the identification, a development or procuring process should begin.

Each item of support and test equipment that was identified by the LSA as a requirement must be a part of a development or a procurement process. The responsible department for support and test equipment should be triggered by status information from the LSA. Nevertheless, requirements for support/test equipment can also be derived from sources other than the documented LSA tasks (especially for common tools, eg a simple screwdriver need not be identified by a special LSA task).
10.1.4 Training

The training area must be specially considered. Training needs concerning maintenance activities cannot be identified until complete information concerning the required maintenance tasks is available (e.g., details of performance, required support equipment, difficulty and criticality, duration, number of working steps, required personnel). The first step in identifying training requirements should be a Training Needs Analysis (TNA). This can be based on the maintenance tasks identified by the LSA. To help identify the best starting point for training activities, status information should be introduced. Additional information should be taken from the technical documentation in order to support the identification of training needs that are not documented in LSA, and to support the creation of training documents in general.

The process to identify and create training should be harmonized with the customer through an agreement concerning LSA and/or technical documentation that is available. The creation of training equipment can be very cost intensive (e.g., production of training videos, training rigs, training simulators). For this reason, decisions concerning training should be based on information that is reliable.

For the best possible support, it is recommended to document training information that is within the LSA database. The LSA database can contain simple markers such as "training required", skill levels for personnel or more detailed information, if available. In addition to these data elements, all possible IT support is recommended to be used to translate training requirements into corresponding queries within the LSA database system.
10.2 Management of the creation of the logistic products

One central challenge within the development or introduction of a new product is the coordination of the product support activities. Especially in case of a new development, the creation of the ILS products is complex. For this reason, it must be ensured that the management of this creation process works properly. It is recommended for ILS managers to use LSA information and the LSA database as a central management tool for the entire ILS process. The following aspects must be considered by the contractor:

- Timely creation of the ILS products must be supported by LSA status information. Triggering the logistic disciplines should be carried out by the support engineering department to ensure a proper start.
- Creating unnecessary effort in any logistic discipline must be avoided, such as:
  - Creation of technical documentation for maintenance actions that are never performed at the customer operational site.
  - Documentation or provisioning of spare parts or consumables that are never required at the customer operational site.
  - Beginning the development or procurement of support/test equipment that are never required at the customer operational site.
  - Planning training for maintenance tasks which are never performed at customer operational site.

A permanent quality check of the content of the logistic disciplines against the content of the LSA database is recommended to avoid effects described in the list above.

10.3 Influence of modifications on logistic products

Design modifications that are relevant for any logistic analysis activity and are relevant for the ILS disciplines must be considered carefully. Logistic disciplines must be triggered in case of changes in the LSA database and, of course, must be triggered in case of design changes with no connected information within the LSA database. The ILS process, as a common process...
from the design to the logistic products, must be carefully managed. It is recommended to establish a strategy for every project phase, on how to manage modifications. Experience shows that even in the design & development phase, many of modifications will have an effect on the entire project. Also, in later phases of the project, modification management for the logistic support products is of vital importance for both the customer and the contractor.

**Fig 27 Influence of design modifications on logistic disciplines in general**

### 11 Checklists

#### 11.1 Detailed checklist for the creation of an ORD
To support logisticians in creating a complete ORD, the following checklist with a number of detailed questions should be used. Nevertheless, it must be clear that special project aspects should be taken into consideration in addition to this list.

**Table 23 Checklist of detailed questions to support the ORD creation**

<table>
<thead>
<tr>
<th>General ORD questions</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which usage areas or which mission areas are expected?</td>
<td>Describe usage areas</td>
</tr>
<tr>
<td>Is the normal usage of the product the only scenario?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Are there special usage scenarios, which must be taken into consideration?</td>
<td>Yes/No? If yes, describe special usage scenarios in detail</td>
</tr>
<tr>
<td>Detailed ORD questions</td>
<td>Answer/Action</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Is a permanent usage expected?</td>
<td>Yes/No? If yes, describe in detail</td>
</tr>
<tr>
<td>Is an out of area usage under rough conditions expected?</td>
<td>Yes/No? If yes, describe in detail</td>
</tr>
<tr>
<td>Is there any influence on the environment expected from noise from the usage or the maintenance of the product?</td>
<td>Yes/No? If yes, describe in detail</td>
</tr>
<tr>
<td>Is there any influence on the environment expected by pollution from the usage or the maintenance of the product?</td>
<td>Yes/No? If yes, describe in detail</td>
</tr>
<tr>
<td>Is there any other influence on the environment expected from the usage or the maintenance of the product?</td>
<td>Yes/No? If yes, describe in detail</td>
</tr>
<tr>
<td>Are there any actions necessary to avoid negative influence on the environment such as noise or pollution?</td>
<td>Yes/No? If yes, describe in detail</td>
</tr>
<tr>
<td>Are there any problems with the currently used product concerning performance?</td>
<td>Yes/No? If yes, describe problems</td>
</tr>
<tr>
<td>Are there any problems with the currently used product concerning maintenance?</td>
<td>Yes/No? If yes, describe problems</td>
</tr>
<tr>
<td>Are there any problems with the currently used product concerning supportability?</td>
<td>Yes/No? If yes, describe problems</td>
</tr>
<tr>
<td>Is the system used in combination with other existing products (generally or only sometimes)?</td>
<td>Yes/No? If yes, describe interaction</td>
</tr>
<tr>
<td>Is the usage of the product dependent on other existing products?</td>
<td>Yes/No? If yes, describe dependence</td>
</tr>
<tr>
<td>Is the product itself transportable?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>How long does it take to prepare the system for transportation?</td>
<td>Duration</td>
</tr>
<tr>
<td>Does transportation require a special conservation and packaging?</td>
<td>Yes/No? If yes, describe in detail</td>
</tr>
<tr>
<td>Define transportability requirements (mode, type, quantity, distance, duration of transport)</td>
<td>Define details</td>
</tr>
<tr>
<td>Which requirements concerning support equipment and personnel exist for the preparation of the product for transport?</td>
<td>Define details</td>
</tr>
<tr>
<td>What is the maximum time to reset the product to full capability after transportation?</td>
<td>Define duration</td>
</tr>
<tr>
<td>Which requirements concerning support equipment and personnel exist for the rest of the product to full capability after transportation?</td>
<td>Define details</td>
</tr>
<tr>
<td>What else must be transported (eg support equipment, spares, personnel) for the proper operation of the system at the destination and what are the requirements for these additional transportation actions.</td>
<td>Define details</td>
</tr>
</tbody>
</table>

**Deployment of locations and special conditions of every location**

| How many operating locations exist? | Number |

Applicable to: All

S3000L-A-03-00-0000-00A-040A-A

Chap 3

2010-04-01 Page 83
<table>
<thead>
<tr>
<th>Detailed ORD questions</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is there an international or even an intercontinental distribution of the operational locations?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>What are the distances between the operating locations?</td>
<td>Map, distances</td>
</tr>
<tr>
<td>What are the distances between the operating locations and the required industry and/or contractor facilities?</td>
<td>Map, distances</td>
</tr>
<tr>
<td>Define the type of each location:</td>
<td>Define details</td>
</tr>
<tr>
<td>Is the location land based?</td>
<td></td>
</tr>
<tr>
<td>Is the location ship based or sea based?</td>
<td></td>
</tr>
<tr>
<td>Is the location based in a mountain region?</td>
<td></td>
</tr>
<tr>
<td>Is the location a depot with maintenance abilities?</td>
<td></td>
</tr>
<tr>
<td>Is the location a home base or an offshore location for the system?</td>
<td></td>
</tr>
<tr>
<td>Is the location a training base equipped with training facilities?</td>
<td></td>
</tr>
<tr>
<td>Define the special conditions at each location:</td>
<td>Define details</td>
</tr>
<tr>
<td>Are there extreme sandy or dusty conditions?</td>
<td></td>
</tr>
<tr>
<td>Is there a salty atmosphere because of location?</td>
<td></td>
</tr>
<tr>
<td>Are there extreme hot or cold weather conditions?</td>
<td></td>
</tr>
<tr>
<td>Are there other extreme stress conditions at special locations?</td>
<td></td>
</tr>
<tr>
<td>Are there limitations due to special regulations/laws at special locations?</td>
<td></td>
</tr>
<tr>
<td>Are there special storage requirements due to extreme weather conditions at special locations?</td>
<td></td>
</tr>
<tr>
<td>In a wartime scenario, are there special locations that would preclude contractor maintenance? (maintenance and repairs in wartime environment cannot easily be performed by contractors)</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Are there special threat situations that might require a special emergency maintenance concept limited to a minimum of intervention?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Is there sufficient infrastructure available to reach all locations (e.g., to guarantee proper supply of spare parts or consumables)</td>
<td>Define details</td>
</tr>
<tr>
<td>Quality and existence of roads</td>
<td></td>
</tr>
<tr>
<td>Airport (local or international) nearby</td>
<td></td>
</tr>
<tr>
<td>Connection to railway stations</td>
<td></td>
</tr>
<tr>
<td>Connection to inland or sea ports</td>
<td></td>
</tr>
<tr>
<td>Are there special infrastructural requirements for reaching a location?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>If yes, define details</td>
<td></td>
</tr>
<tr>
<td>What are the existing capabilities of each location concerning support equipment?</td>
<td>Define details</td>
</tr>
<tr>
<td>What are the existing capabilities of each location concerning infrastructure and facilities?</td>
<td>Define details</td>
</tr>
<tr>
<td>What are the existing capabilities of each location concerning personnel?</td>
<td>Define details</td>
</tr>
<tr>
<td>What are the planned capabilities of each location concerning support equipment?</td>
<td>Define details</td>
</tr>
<tr>
<td>What are the planned capabilities of each location concerning infrastructure and facilities?</td>
<td>Define details</td>
</tr>
</tbody>
</table>
### Detailed ORD questions

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the planned capabilities of each location concerning personnel?</td>
<td>Define details</td>
</tr>
<tr>
<td>Is it planned to establish a repair station at special locations?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Is it planned to establish a supply depot at special locations?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Is it planned to share abilities between different locations (exchange of support equipment or personnel)</td>
<td>Yes/No?</td>
</tr>
</tbody>
</table>

### Products (systems) supported and product (system) deployment

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many supported products are planned per location?</td>
<td>Define details</td>
</tr>
<tr>
<td>Deployment of products (end items) per location</td>
<td>Define details</td>
</tr>
<tr>
<td>Is it planned to share products (end items) between different locations?</td>
<td>Define details</td>
</tr>
</tbody>
</table>

### Detailed usage overview

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>What will be the normal usage of the product at every location (key usage)?</td>
<td>Define details</td>
</tr>
<tr>
<td>What is the planned availability of the product at each operating location?</td>
<td>Define details</td>
</tr>
<tr>
<td>When the product is used in missions, what are the planned mission success rates for each operating location?</td>
<td>Define details</td>
</tr>
<tr>
<td>Are there any periods of special usage of the products at a special location?</td>
<td>Define details</td>
</tr>
<tr>
<td>low payload period</td>
<td></td>
</tr>
<tr>
<td>temporary peak loads</td>
<td></td>
</tr>
<tr>
<td>temporary storage of the product</td>
<td></td>
</tr>
<tr>
<td>out of area usage under special or rough conditions</td>
<td></td>
</tr>
<tr>
<td>wartime usage</td>
<td></td>
</tr>
<tr>
<td>In the case of permanent operational conditions, what are the possible down times for maintenance (maintenance windows)?</td>
<td>Define time periods for maintenance</td>
</tr>
<tr>
<td>Define additional system performance parameters in measurable quantifiable terms such as:</td>
<td>Define detailed values</td>
</tr>
<tr>
<td>range of usage</td>
<td></td>
</tr>
<tr>
<td>required accuracy</td>
<td></td>
</tr>
<tr>
<td>payload values</td>
<td></td>
</tr>
<tr>
<td>required temperatures</td>
<td></td>
</tr>
<tr>
<td>required speed</td>
<td></td>
</tr>
<tr>
<td>required distances</td>
<td></td>
</tr>
<tr>
<td>What is the key measurement base of usage per unit of time? Examples can be the following:</td>
<td>Define detailed values</td>
</tr>
<tr>
<td>Operational hours for general products</td>
<td></td>
</tr>
<tr>
<td>Flight hours for airborne products (aircrafts, helicopters)</td>
<td></td>
</tr>
<tr>
<td>Kilometers or miles for land based vehicles</td>
<td></td>
</tr>
<tr>
<td>Rounds or cycles for periodic processes in a product</td>
<td></td>
</tr>
<tr>
<td>Tons for transportation systems</td>
<td></td>
</tr>
</tbody>
</table>

| defined values |
Detailed ORD questions

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>What operational profile is expected at each location?</td>
<td>Define a operational profile as detailed as possible</td>
</tr>
<tr>
<td>How many operating hours or missions are planned per day?</td>
<td>Define detailed values</td>
</tr>
<tr>
<td>Are there different typical usage days, define the operating profile for each “day type”.</td>
<td></td>
</tr>
<tr>
<td>What are the typical operational profiles for longer ranges such as weeks, months or entire years?</td>
<td></td>
</tr>
<tr>
<td>How many operating days/operating hours are expected for a year or for any other basic time period?</td>
<td>Define detailed values</td>
</tr>
<tr>
<td>What is the average duration of each different usage (eg operation, mission, trip, flight, dive)</td>
<td>Define detailed values</td>
</tr>
</tbody>
</table>

11.2 Detailed checklist for the creation of an CRD

To support logisticians to create a complete CRD, the following checklist with a number of detailed questions should be used. Nevertheless, it must be clear that special project aspects should be taken into consideration in addition to this list.

Table 24 Checklist of detailed questions to support the CRD creation

<table>
<thead>
<tr>
<th>Supply concept</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are central depots expected, and how is the deployment planned?</td>
<td>Define details</td>
</tr>
<tr>
<td>Are depots planned directly at the operational locations?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Is a material exchange between different locations expected?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Is a selection process for suppliers defined and how is it planned to integrate suppliers into the expected supply concept?</td>
<td>Define details</td>
</tr>
<tr>
<td>Is it possible for a production line to provide spare part support (especially at early stages)?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Is an outsourcing concept via a support agency expected?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>What are the limitations for spare part availability and lead times?</td>
<td>Define details</td>
</tr>
<tr>
<td>What distribution systems and IT-system shall be used for spare part supply?</td>
<td>Define details</td>
</tr>
<tr>
<td>Which concept shall be used for initial provisioning?</td>
<td>Define details</td>
</tr>
<tr>
<td>Is it expected to use an optimization or a simulation tool?</td>
<td></td>
</tr>
<tr>
<td>Is the deployment schedule properly supported by the initial spares delivery?</td>
<td></td>
</tr>
<tr>
<td>Will spare part delivery impact the production schedule?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Will the industrial base be obligated to provide spares support in the out-years for items that remain in the inventory?</td>
<td>Yes/No?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Support equipment concept</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>How can it be ensured that all support equipment is available in time?</td>
<td>Define details</td>
</tr>
</tbody>
</table>
### Detailed CRD questions

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the identification and realization process for standard and special support equipment?</td>
<td>Define details</td>
</tr>
<tr>
<td>How effectively is automated test equipment being utilized to support the product?</td>
<td>Define details</td>
</tr>
<tr>
<td>What is planned for maintenance and support of the support equipment itself?</td>
<td>Define details</td>
</tr>
<tr>
<td>Repair and overhaul of support equipment</td>
<td></td>
</tr>
<tr>
<td>Calibration of support equipment</td>
<td></td>
</tr>
</tbody>
</table>

#### Personnel integration

- **What are the general requirements and provisions for manpower and personnel?** Define details
- **Are the products operated by the staff of the customer or by staff of the contractor?** Define details
- **Should cooperative models be considered?** Yes/No?
- **How can it be ensured that the required operator training is verified on time?** Define details
- **How can it be ensured that the required maintenance training is verified on time?** Define details
- **What training processes should be developed to ensure adequate operational and maintenance support at all levels during the entire life of the system? The possible rapid turn-over of personnel should especially be considered.** Define details

#### Facilities

- **Which services must be available at the required facility (e.g., electrical power, hydraulic power, compressed air, special working environment)?** Define details
- **Are existing facilities at operational locations appropriate to the requirements of a new product (operational and maintenance facilities)?** Yes/No?
- **Can existing facilities at operational locations be adapted to the requirements of a new product (operational and maintenance facilities)?** Yes/No?
- **Is there a realistic time schedule available for the building of required new facilities?** Yes/No? Define details
- **Is there a plan to identify and reduce risks caused by the delay of the building of facilities?** Define details

#### IT and communication resources

- **Which IT architectures exist at every operational location concerning the following aspects:** Define details
- Network capabilities
- Data storage capabilities
- Computer capabilities
- Communication resources
<table>
<thead>
<tr>
<th>Detailed CRD questions</th>
<th>Answer/Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are the existing IT and communication capabilities appropriate for the new products to be introduced?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Are there risks in connection with future changes in IT or communication architecture concerning the operation of the new products?</td>
<td>Define details</td>
</tr>
<tr>
<td>Are there concepts for an initial provision with data for required IT systems?</td>
<td>Define details</td>
</tr>
<tr>
<td>Who is the holder of the required data?</td>
<td></td>
</tr>
<tr>
<td>Are there contractual aspects to be considered?</td>
<td></td>
</tr>
<tr>
<td>Are there any problems concerning data security requirements?</td>
<td></td>
</tr>
<tr>
<td>Are there plans for a first installation of required new IT systems and for the appropriate service of IT systems during the life of the product to be supported?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Are there service contracts for hardware and software (planned and existing)?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Fast reaction to downtime must be ensured. Upgrade of hardware and software in case of technical progress should be included in the agreements of a service contract.</td>
<td></td>
</tr>
<tr>
<td>Are different IT systems at different operational locations compatible?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td>Is there a plan to ensure the existence of all required interfaces to other IT systems?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td><strong>New organizational structures</strong></td>
<td></td>
</tr>
<tr>
<td>Will any organizational structures be changed at operational locations and what are the risks of these changes?</td>
<td>Define details</td>
</tr>
<tr>
<td>What structure changing opportunities exist with the introduction of a new product (eg reduction of personnel)</td>
<td>Define details</td>
</tr>
<tr>
<td><strong>Schedule consideration</strong></td>
<td></td>
</tr>
<tr>
<td>What are the plans for an appropriate participation of logistic disciplines in all phases of the program?</td>
<td>Define details</td>
</tr>
<tr>
<td>Are the logisticians involved in the program in a very early phase of the program so they can influence basic decisions concerning design?</td>
<td>Yes/No?</td>
</tr>
<tr>
<td><strong>Additional aspects</strong></td>
<td></td>
</tr>
<tr>
<td>What additional aspects specific to the project are of high significance for the logisticians?</td>
<td>Define details</td>
</tr>
</tbody>
</table>

11.3 **Breakdown element identifier - structure examples**

The following examples show how an effective product breakdown could be established. For this purpose, examples from Para 5.2 are used and filled with the missing BEIs for each breakdown level and for each breakdown element. Additionally, examples of how to integrate software items are part of this paragraph.
11.3.1 **Functional breakdown**

The first example (Fig 28) shows a functional breakdown following a simple BEI syntax structure. The top level item (root) can be a product containing a fuel system.

![Functional breakdown, simple syntax with separators](ICN-B6865-S3000L0027-001-01)

**Table 25** Listing of functional breakdown from Fig 28

<table>
<thead>
<tr>
<th>BEI</th>
<th>Item name</th>
<th>Item type</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>Fuel system</td>
<td>Main system</td>
</tr>
<tr>
<td>28-01</td>
<td>Fuel storage</td>
<td>Function</td>
</tr>
<tr>
<td>28-04</td>
<td>Fuel distribution</td>
<td>Function</td>
</tr>
<tr>
<td>28-04-01</td>
<td>Tank interconnection</td>
<td>Subfunction</td>
</tr>
<tr>
<td>28-04-02</td>
<td>Fuel transport to engine</td>
<td>Subfunction</td>
</tr>
<tr>
<td>28-04-03</td>
<td>Refueling</td>
<td>Subfunction</td>
</tr>
<tr>
<td>28-04-03-01</td>
<td>Pressure refueling</td>
<td>Sub subfunction</td>
</tr>
<tr>
<td>28-04-03-02</td>
<td>Gravity refueling</td>
<td>Sub subfunction</td>
</tr>
<tr>
<td>28-05</td>
<td>Fuel injection</td>
<td>Function</td>
</tr>
<tr>
<td>28-06</td>
<td>Control &amp; indication</td>
<td>Function</td>
</tr>
</tbody>
</table>

The second example (Fig 29) shows a functional breakdown with an extended BEI syntax structure. The top level item (root) can be any vehicle or product containing a fuel system; the first function/system is the fuel system itself. The BEI includes information about the end item itself (V125) and about the BEI type (F=functional). In this case, the fuel system is the first level of the breakdown tree.

**Note**

For a common understanding, it should be noted that the end item within a tree should be considered as level zero, or in other words, as the level "root". The next level down from the root should be the first breakdown level.
11.3.2 Mixture of functional and physical breakdown

The next example (Fig 30) shows a typical mixture of a functional and a physical breakdown. Within this breakdown, starting from the root level, the first breakdown levels are used for functional/system information. From a certain depth of breakdown downwards you will find real physical equipment and parts. The functional information will be fixed inside the breakdown. This means that a functional system can be identified by the first breakdown level of the BEI, a functional subsystem can be identified by the second breakdown level and so on. Within complex products, systems can be of different size. For that reason normally an indenture level where functional breakdown ends and physical breakdown starts cannot be fixed. It is recommended to use existing SNS (eg chapters of the S1000D) for an adequate structuring of the product under analysis.
As shown in Fig 30, sometimes the border between a functional BEI and a physical BEI is fluid. For example the BEI representing the internal storage tanks in Fig 30 could be a functional BEI, covering the function internal fuel storage, or could be an assembly of physical components, in this case, the four internal tanks.

11.3.3 Separation of physical and functional breakdown with cross referencing (hardware only)
This example (Fig 31) shows the usage of totally separate breakdown trees for both functional and physical breakdowns. The problem of finding an effective connection between these two areas should be solved within a LSA database by the means of appropriate cross reference tables.
The cross referencing between separate breakdowns should be realized in a $m:n$ relationship structure. On the one hand, one special function can be connected to many ($m$) hardware components, on the other hand one special hardware component can be connected to more than one ($n$) function. In this context, this kind of connection is called a $m:n$ relationship between the functional and the physical breakdowns.
11.3.4 Physical breakdown with explicit parent-child relationship

Another philosophy of breakdown is the usage of a parent-child methodology. In this case, the hardware is organized in a hierarchical way concerning the installation of the equipments or components in the corresponding parent breakdown element. This type of breakdown is used, for example, in PDM systems. Additional information in form of a BEI can also be given. The functional entity (as given in the examples before by using the "28" from a SNS for fuel system) is not automatically given in this type of breakdown methodology.

![Diagram of parent-child breakdown]

In this type of breakdown, as shown in Fig 33, the key information given by, for example a part number of the BE, does not contain any information concerning the functional entity. The interrelation between the BE’s not given by the key data element part number. Using this type of breakdown methodology requires the information about the next higher BEI to be able to create the tree structure of the complete breakdown. The information given in the following table must be documented in a corresponding LSA database system when using the parent-child methodology.

<table>
<thead>
<tr>
<th>Item number</th>
<th>Item name</th>
<th>Item type</th>
<th>Next higher item</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Vehicle V1</td>
<td>Product</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FT002-791</td>
<td>Fuel tank</td>
<td>Assembly</td>
<td>V1</td>
<td>3</td>
</tr>
<tr>
<td>FT102-591-A</td>
<td>Auxiliary tank</td>
<td>Assembly</td>
<td>V1</td>
<td>2</td>
</tr>
<tr>
<td>ATF-009-G-21</td>
<td>Rubber tank</td>
<td>Component</td>
<td>FT002-791</td>
<td>1</td>
</tr>
<tr>
<td>G0135991</td>
<td>Cover tank</td>
<td>Component</td>
<td>FT002-791</td>
<td>1</td>
</tr>
<tr>
<td>VG7-1206</td>
<td>Drain valve</td>
<td>Component</td>
<td>FT002-791</td>
<td>1</td>
</tr>
<tr>
<td>G05H32110</td>
<td>Blanking plug</td>
<td>Component</td>
<td>FT002-791</td>
<td>1</td>
</tr>
<tr>
<td>H63433-A</td>
<td>Filter</td>
<td>Component</td>
<td>VG7-1206</td>
<td>1</td>
</tr>
<tr>
<td>1205664</td>
<td>O-Ring</td>
<td>Component</td>
<td>H63433-A</td>
<td>1</td>
</tr>
<tr>
<td>1205667</td>
<td>O-Ring</td>
<td>Component</td>
<td>H63433-A</td>
<td>1</td>
</tr>
</tbody>
</table>
Within an explicit breakdown, it is also possible to document multiple installations of components. In the case that there is a need to be explicit about each installation location, each installation location has to be documented as a separate BE with quantity 1.

11.3.5 **Integration of software as part of a single physical hardware item**

The first case to be examined is the ability to assign a SW package directly to one piece of hardware equipment. The assignment of a BEI for the SW can be done at the level below the hardware according to the established breakdown coding rules.

The coding rules can include special digits to identify the BEI as a hardware BEI or a SW BEI. In breakdown example 1 of [Fig 34](Fig), the BEI of the Navigation radar and of the Navigation SW package embedded in the Navigation radar cannot be separated based on whether it is a hardware or a SW BEI. But it is clear that the SW package can be assigned uniquely to the Navigation radar as the carrier of the SW.

![Breakdown example 1](Breakdown example 1)

In breakdown example 2 of [Fig 34](Fig), the same breakdown is shown, but with a "software BEI". It is marked as a SW BEI using the two digits "SW" within the BEI. The other breakdown levels are not affected, and the assignment of the SW package as a part of the Navigation radar is unique.

11.3.6 **Integration of software as part of more than one physical hardware items**

This situation is more difficult concerning the assignment of hardware. If the installation or loading of SW distributes a software package to more than one physical device in the hardware breakdown, a unique assignment of SW to hardware is not possible. An alternative method to assign the SW is to shift the SW component to a higher breakdown level. This is shown in [Fig 35](Fig).

![Breakdown example 2 (with software BEI)](Breakdown example 2 (with software BEI))

**Fig 34** Assigning of software BEI in a physical breakdown

In breakdown example 2 of [Fig 34](Fig), the same breakdown is shown, but with a "software BEI". It is marked as a SW BEI using the two digits "SW" within the BEI. The other breakdown levels are not affected, and the assignment of the SW package as a part of the Navigation radar is unique.
Fig 35  Distributed software in a physical breakdown

The loadable radar SW in Fig 35 can be loaded in one procedure to two different pieces of equipment. The SW cannot be divided up physically into two parts, because it is delivered and installed as one single object. The dividing is done by the loading procedure; a unique assignment is not possible.

In this case, it is possible to address the SW component one level higher to the BEI of the complete radar system. In order to have the correct cross referencing to the appropriate hardware components, it may be necessary to establish a cross reference assigning the BEI of the SW package to both the Navigation radar and to the Weather radar. This would be important if the radar system contains further hardware equipment that are not loading devices for the SW package.

11.3.7 Integration of software in a functional breakdown

In a functional structure, SW items should be shown as children of the functional system or breakdown element within which they operate. In allocating a BEI, it is necessary to consider the supportability characteristics of the SW items. Items with particular support requirements should be allocated to their own BEI. For example, the BEI scheme should distinguish between SW that is loadable and SW that is embedded. It may also be necessary to distinguish between SW that has been provided by the prime contractor or SW items that are safety or security significant. During the early stages of a project these supportability issues might not have been fully considered. The development of a functional BEI allocation scheme should be seen as an iterative process.

In order to develop a functional BEI structure in which SW items are logically grouped at an appropriate level of indenture, and in which the differing support requirements of all software items are exposed, items will often be included at levels of indenture against which only limited amounts of information needs to be stored. For instance, in the example below (Fig 36), little LSA data will be stored in respect to items Fourier and Signature. The full range of LSA database information will be recorded against the items at the lowest indenture level. In such a case, SW is represented by the entities on which the software modification activity is carried out.
It is necessary to maintain both functional and physical views of SW items throughout the life of the parent product. It can be important to establish links between software items in the functional domain and the associated executable SW package in the physical domain. This may be achieved by the same method of cross mapping the physical hardware items to functional product breakdown.

The requirements concerning the assignment of SW BEI in a pure functional breakdown are the same as in physical breakdowns. If there are functions that are realized by exactly one definite SW package, the unique assignment would be possible. If more than one function is realized by the loading of a SW package, the need for a higher level assignment is the same as for the physical breakdown methodology.

An additional question concerning SW is when to use a functional breakdown and when to use a physical breakdown. In many cases, SW can be assigned within a physical breakdown on a subsystem or system level (this can be an additional argument to use a mixed breakdown methodology, physical and functional). However, for SW modification purposes, a functional breakdown is preferred within a separated SSA database. Normally, a mixed breakdown structure including the higher breakdown hierarchies on system/subsystem level as a kind of functional framework is not sufficient for a real functional breakdown. On the other hand, a complete mixture of functional and physical breakdown aspects within one breakdown is not feasible.
11.4 LSA candidate selection flowchart examples

11.4.1 Non structural items

![Flowchart Diagram]

**Fig 37 Full LSA candidate selection flowchart**

The selection flowchart from **Fig 37** shows a typical process for the determination of full LSA candidates from an existing breakdown. The flowchart must be applied to each breakdown element. If the selection flowchart for full LSA candidates leads to the IUA not being a full LSA.
Candidate, the second LSA selection flowchart for partial candidates must be applied to each of those BE’s.

**Fig 38 Partial LSA candidate selection flowchart**

Depending on the project, the selection criteria must be adapted since additional criteria may be required. In general, the analyst must have a guideline for the LSA candidate selection. The examples here are simple and clear. Any required change of flowchart logic within a special project must be harmonized between the contractor and the customer and accepted by the customer during the LSA GC.

### 11.4.2 Structural items

In the case of structural components, the selection of LSA candidates differs from the selection process for non-structural items.

---

**ICN-B6865-S3000L0037-001-01**
Fig 39  LSA candidate selection flowchart for structural items
# Configuration management

## Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration management</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>4</td>
</tr>
<tr>
<td>2 Configuration management in the LSA process</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Configuration management planning</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1 Milestone plan</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2 Milestone input criterion (trigger)</td>
<td>5</td>
</tr>
<tr>
<td>2.1.3 Milestone output criterion (trigger)</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Configuration identification</td>
<td>5</td>
</tr>
<tr>
<td>2.2.1 General aspects</td>
<td>5</td>
</tr>
<tr>
<td>2.2.2 Configuration identification rules (steps)</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Configuration control</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Configuration accounting</td>
<td>7</td>
</tr>
<tr>
<td>2.5 Configuration verification and audit</td>
<td>7</td>
</tr>
<tr>
<td>2.5.1 Functional configuration audit</td>
<td>7</td>
</tr>
<tr>
<td>2.5.2 Physical configuration audit</td>
<td>7</td>
</tr>
<tr>
<td>2.5.3 Configuration audit requirements</td>
<td>8</td>
</tr>
<tr>
<td>3 Sources/drivers for configuration changes</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Customer requirements for change</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Design change proposals</td>
<td>9</td>
</tr>
<tr>
<td>3.3 Manufacturing deviation from design (concessions)</td>
<td>10</td>
</tr>
<tr>
<td>3.4 Supplier changes</td>
<td>10</td>
</tr>
<tr>
<td>3.5 Inservice changes (service bulletins)</td>
<td>10</td>
</tr>
<tr>
<td>4 The impact of configuration changes in LSA</td>
<td>10</td>
</tr>
<tr>
<td>4.1 Evaluation of the impact of configuration changes</td>
<td>10</td>
</tr>
<tr>
<td>4.2 Traceability between the source of changes and consequential LSA changes</td>
<td>11</td>
</tr>
<tr>
<td>5 Processing of configuration changes during the LSA process</td>
<td>11</td>
</tr>
<tr>
<td>5.1 General aspects</td>
<td>11</td>
</tr>
<tr>
<td>5.2 Coordinating role of LSA</td>
<td>11</td>
</tr>
<tr>
<td>5.3 Task assignment by discipline</td>
<td>12</td>
</tr>
<tr>
<td>5.4 Task status and control</td>
<td>12</td>
</tr>
<tr>
<td>6 Implementation of configuration management</td>
<td>13</td>
</tr>
<tr>
<td>6.1 Change compilation and traceability</td>
<td>13</td>
</tr>
<tr>
<td>6.2 LSA configuration items identification</td>
<td>13</td>
</tr>
<tr>
<td>6.3 Support system and other ILS configuration items definitions</td>
<td>13</td>
</tr>
<tr>
<td>6.3.1 Technical documents configuration items</td>
<td>14</td>
</tr>
<tr>
<td>6.3.2 Illustrated parts list and illustrated parts catalogue configuration items</td>
<td>14</td>
</tr>
<tr>
<td>6.3.3 Support equipment configuration items</td>
<td>14</td>
</tr>
<tr>
<td>6.4 LSA compilation by change</td>
<td>15</td>
</tr>
<tr>
<td>6.4.1 General</td>
<td>15</td>
</tr>
<tr>
<td>6.4.2 Operational system configuration compilation</td>
<td>16</td>
</tr>
<tr>
<td>6.4.3 Functions and descriptions configuration</td>
<td>17</td>
</tr>
<tr>
<td>6.4.4 Failure modes and effects analysis configuration</td>
<td>17</td>
</tr>
<tr>
<td>6.4.5 Task analysis configuration</td>
<td>18</td>
</tr>
<tr>
<td>6.4.6 Scheduled Maintenance Analysis configuration management</td>
<td>19</td>
</tr>
</tbody>
</table>
6.4.7 Support equipment change compilation ............................................................. 19
6.5 Applicability management .................................................................................... 20
6.5.1 General ............................................................................................................... 20
6.5.2 Manufactured serial number and fleet serial number concepts ......................... 20
6.5.3 Effectivity of change/criterion ........................................................................... 22
6.5.4 MSN versus FSN applicability management ...................................................... 23
6.5.5 Product variant management .............................................................................. 23
6.5.6 Serialized item management .............................................................................. 24
6.6 Production and releases management .................................................................... 24
6.6.1 General ............................................................................................................... 24
6.6.2 End product identification .................................................................................. 25
6.6.3 Change traceability of releases ......................................................................... 25
6.6.4 Release management ......................................................................................... 26
6.6.5 Customization of a release ................................................................................ 27

List of tables

1 References ................................................................................................................. 2
2 List of configuration identification rules (steps)......................................................... 6
3 Types of affectation .................................................................................................... 15

List of figures

1 Configuration management process ......................................................................... 3
2 Configuration traceability from design configuration to ILS/LSA configuration ........... 13
3 LSA compilation by change ....................................................................................... 16
4 Applicability calculation based in the criteria applicability ......................................... 20
5 MSN usage example .................................................................................................. 21
6 Fleet example ............................................................................................................. 21
7 FSN (fleet) versus MSN ............................................................................................ 22
8 Applicability calculation ............................................................................................ 23
9 Example for applicability to Product versions ............................................................ 24
10 Release management ................................................................................................ 27

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 3</td>
<td>LSA Business process</td>
</tr>
<tr>
<td>S1000D</td>
<td>International specification for technical publications using a common source database</td>
</tr>
<tr>
<td>S2000M</td>
<td>S2000M - International specification for material management</td>
</tr>
</tbody>
</table>

Applicable to: All

S3000L-A-04-00-0000-00A-040A-A

Chap 4

DMC-S3000L-A-04-00-0000-00A-040A-A_001_00_EN-US.doc

2010-04-01 Page 2
1 General

1.1 Introduction

Configuration Management (CM) is a discipline that ensures the correct identification of the configuration, controls changes, and records the change implementation status of the physical and functional characteristics of the structure, system/subsystem and equipment.

CM is to be performed at the lowest level of the breakdown of a product. It identifies what is required, designed, produced, supported and evaluates changes, including effects on technical and operational performance and Integrated Logistic Support (ILS) activities. It also provides visibility of the configuration.

The elements of configuration management include:
- Configuration management planning
- Configuration identification
- Configuration control
- Configuration accounting
- Configuration verification and audit

CM is therefore the means through which integrity and continuity of the design, systems engineering and supportability are recorded, communicated, and controlled. CM efforts result in the complete audit traceability of decisions and design modifications. Suitably qualified contractor and customer personnel, who will certify the components and inform CM of the outcome, will perform these efforts. The following figure shows a schematic of how the entire process of CM works.

**Fig 1 Configuration management process**

ICN-B6865-S3000L0076-002-01
Explanation of the process steps in Fig 1:

1. Any external change when impact becomes criteria for change in LSA.
2. As a consequence of a criterion, a set of activities need to be performed so planning and control for them is required.
3. Criteria are used to authorize change in the LSA products and also guarantee the traceability of changes internally and externally to the sources.
4. The applicability of criteria + impact/authorization of criteria over the LSA configuration items is used in the applicability management to provide the applicability of the LSA configuration Items.
5. Provides the status of the criteria included in the set that defines a delivery.
6. Provide applicability to all components of the LSA products.
7. Provides contents for the LSA products and the criteria traceability to the LSA products.
8. Provides milestones for deliveries or releases of the LSA products.

1.2 Objective
The goal of this chapter is to present all processes involved in the CM for LSA guidelines and examples are given in this chapter to allow efficient CM of the operational system as well as the support system. This chapter covers all aspects of CM whether or not they are used for a specific project. Users of this chapter must tailor the process to their needs.

1.3 Scope
This chapter is focused on the LSA CM, but since LSA is a core ILS process, it has many relationships with other CM in other ILS disciplines. Most of the guidelines and rules given in this chapter can be applied to other ILS disciplines and this chapter strongly relates to other ASD specifications.

This chapter is oriented toward all LSA and ILS managers and it is recommended that this chapter is also read by other ILS discipline managers.

2 Configuration management in the LSA process
2.1 Configuration management planning
CM planning in the LSA process shall start at the beginning of the program and continue through all phases of the Product’s life cycle. CM planning results in an LSA CM Program (LCMP). The LCMP must account for the customer requirements regarding CM. The LCMP shall identify a series of milestones/meetings, which are documented in a milestone plan.

An LCMP shall comply with the following:
- Outline and define the organizational structure for developing the project (Service).
- Define policies and methods used to establish and control the baselines and configuration items identified.
- Describe the plans for the application of configuration index, matrix and change status accounting records and procedures.
- Describe methods for meeting CM requirements for continuous acquisition and life cycle support.
- Define the proposed methods for control and monitoring of any subcontractors and vendors.
- Establish the major milestones for implementation of configuration management.
- Describe integration of CM activities with other program/management activities.
Describe the plans for conducting/supporting the functional and physical configuration audits at any product management level (system/subsystem, equipment/module level).

2.1.1 Milestone plan
The milestone plan must generally follow the program/design milestones. However, it is important to establish a buffer between the design milestone and the relevant LSA milestone in order to leave enough time to allow ILS personnel to incorporate the design changes. This milestone plan requires an appropriate number of meetings between the customer and the contractor for approval of the milestone accomplishment.

2.1.2 Milestone input criterion (trigger)
Every milestone will have an input criterion. This is the necessary documentation to be delivered by the contractor/customer to properly conduct the milestone meetings. This input criterion (trigger) must be delivered with enough time prior to the meeting to allow proper analysis and ensure an efficient meeting. Non-compliance of any of this criterion may result in a cancellation of the meeting.

2.1.3 Milestone output criterion (trigger)
These are the results of the milestone meeting and provide the action plan for the next phase until the next milestone meeting.

2.2 Configuration identification
2.2.1 General aspects
Configuration identification consists of setting and maintaining baselines that define the system or subsystem and their development at any point in the life cycle. From the ILS point of view, the system to be considered comprises the mission system (eg aircraft) and the support system. This means that ILS focuses not only on production, but also on servicing the product (mission system).

− The mission system is traditionally considered to be the product. It is the system, along with all of the functions necessary to comply with the established mission, to be operated by the customer.
− The support system includes all elements necessary to support the mission system and to allow it to comply with the mission in the most effective way. All the elements of ILS can be considered as the support system:
  • Support equipment. This includes any tool and support and test equipment necessary to operate/support the mission system.
  • Technical documentation
  • Manpower
  • Training and training equipment
  • Facilities and any other item required in the operation and support of the mission system

Effective configuration identification is a pre-requisite for all other CM activities. If configuration identification and their associated configuration documentation are not properly identified, it is impossible to control the changes to the item’s configuration, to establish accurate records and reports, or to validate the configuration through audits.

Inaccurate or incomplete configuration documentation may result in defective products, schedule delays, and higher maintenance costs after delivery.

2.2.2 Configuration identification rules (steps)
Configuration identification must follow specific rules (steps) that are outlined below.
Table 2 List of configuration identification rules (steps)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification of configuration items (Functional and physical)</td>
<td>The identification of the item must follow the design configuration and be compatible with other ASD specifications like S1000D or S2000M. For example, the Standard Numbering System (SNS) from S1000D could be followed for the functional configuration.</td>
</tr>
<tr>
<td>Relationship between Functional and physical configuration</td>
<td>Functional and physical configurations are two different points of view of the same item. Any item in the functional configuration must be identified in the physical configuration and vice versa, therefore, requirements for the management and link between these configurations must be establish in the LCMP.</td>
</tr>
<tr>
<td>Relationship between design and LSA identification</td>
<td>Rules to guarantee the traceability between design configuration and LSA configuration must be included in the LCMP.</td>
</tr>
<tr>
<td>Zoning and access panel identification</td>
<td>Structuring the product by zones to include identification of access panels is another approach to the configuration of a product.</td>
</tr>
<tr>
<td>Functional Identification Number /Reference designator identification</td>
<td>Identifying an item for schematic purposes (eg electrical) can be accomplished through the Functional Identification Number or Reference designator (according to S2000M). Rules for Functional identification number / Reference designator assignment can be included in the LCMP.</td>
</tr>
<tr>
<td>Variants/Alternate item identification</td>
<td>The use of variant must be defined at the beginning of the project and included in the LCMP. This coding is used for identifying different technical solutions for the same function. The relation to other specifications is also useful, for example, the relation between an alternate breakdown element identifier and an Item sequence number. Chap 3 provides additional information on documenting product variants and alternate items.</td>
</tr>
</tbody>
</table>
| Top level configuration identification | At the beginning of the project, the type of top level configuration management must be established. Top level configuration management serves to identify the products or any other kind of grouping of products. For example, it is necessary to define the use of different top level management:  
- By manufactured serial number  
- By fleet number or customer versions  
- By system difference codes  
- By applicability. This defines a way of managing the configuration by the part number. It is used for items for which serial number management is not suitable. |
| Required documentation for proper configuration identification | The documentation necessary to properly identify the configuration (eg ASD breakdown document, Illustrated Parts Data (IPD)) must be defined at the beginning of a project. |

2.3 Configuration control

Configuration control includes the evaluation of all change requests, change proposals, and their subsequent approval or disapproval.
This shall ensure that no change will be implemented without due consideration of its effect on the baselines, including logistics impact, costs, schedules, performance, or interface with any associate companies, etc.

The authority required to make a decision on a change varies with the magnitude and nature of the change concerned. It involves the appropriate levels within the organization of the customer and the developer for the project. The decision making process is an integral part of the overall management of the "project management system".

To enable the process to operate correctly, it is necessary to establish specific procedures for dealing with changes, for example:

- Providing the relevant information for best decisions on changes
- Implementing decisions
- Reviewing and controlling changes in accordance with established rules

2.4 **Configuration accounting**

The LCMP shall establish necessary procedures/requirements that ensure the correct monitoring of changes and their effect on configuration. These are as follows:

- Up-to-date accounting of the exact configuration of each product
- List of changes status
  - Design changes
  - Waivers
  - Deviations
- Procedure for change status control
- Requirements for version versus changes incorporated
- Discrepancies from audits management requirements

2.5 **Configuration verification and audit**

Functional and physical configuration audits will be performed in accordance with the general and detailed requirements and processes. The main purpose of the Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA) is to establish a product baseline and to establish a system or production baseline. This defines the configuration of the product for manufacture and provides assurance for the support of the product during operational use.

2.5.1 **Functional configuration audit**

The FCA will be an examination of the configuration identification documents to ensure that the technical documentation accurately reflects the actual functional characteristics of a configuration item and conforms to the necessary interface characteristics.

Test and evaluation data will be examined to verify the configuration item has achieved the performance specified in the relevant configuration identification documentation, ie, functional and allocated baselines. The FCA will be performed on the first item that is representative of the configuration which is to be used as a deliverable item. Any difference will be identified between the physical configuration of the selected configuration item and other development configuration items used for the FCA, and will certify or demonstrate that these differences do not degrade the functional characteristics of the selected configuration item.

2.5.2 **Physical configuration audit**

The PCA will be an audit of the configuration identification documentation, planning and manufacturing data, released engineering drawings, and the quality evaluation records to ensure that the configuration baseline engineering documents have been adhered to and that the "as-built" or "as-coded" condition is in accordance with them. The PCA will be performed on the first item used for qualification and compliance verification. A system PCA may be
necessary for large and complex systems to ensure that the physical baseline build standard is acceptable for production release.

2.5.3 Configuration audit requirements
Both the functional and physical audit should meet following requirements:

- Accuracy and acceptability of the functional and product baselines and assure that the Configuration is, in fact, built to these baselines
- Contract requirements
- Audit plan
- Audit procedure
- Items subject to audit
- Audit scheduling and milestones
- Input key
- Output key
- Audit responsibilities

Responsibility for the physical and functional configuration audits is with the CM technical authority and shall be performed by the acquirer or his nominated representative with support from the following:

- System/subsystem design manager
- Quality representative
- Corrective action and risk management requirements

3 Sources/drivers for configuration changes
LSA CM is required because there are changes to the baseline configuration of a project. These changes may be internal changes (e.g., improvement of the support analysis) or external changes (e.g., design changes, supplier changes).

Potential Drivers/Sources for configuration changes may be:

- Requirements from the customer.
- Changes from the design office
- Manufacturing deviation from design (Concessions)
- Supplier changes
- Customer orders for change

Since specific drivers are linked to specific sources, the management of the relevant sources is required for CM. Specific changes normally will follow defined processes and will require specific activities. The most important business cases are outlined below.

3.1 Customer requirements for change
In some cases, the customer requires changes to the end product. The process that needs to be created when the customer requires a change is as follows:

- Reception of Customer Originated Change (COC) proposal
- COC management
  - Analysis of impact with cost implications. The request for change from the customer must be analyzed to determine the impact of the change proposal and the cost to implement the change.
  - Change committee actions. The change can be submitted to the modification committee for approval or rejection.
- Decision and Resolution notification. The decision on the change proposal should be communicated to the customer.
  - COC design validation
    - Requirement for design office to make an analysis. The change proposal is taken to the design office for analysis.
    - Design change proposal associated. When the design office decides to create a design modification, a link between the customer change proposal and the design change must be established and recorded for traceability purposes.
    - Design office documentation collection. The design office prepares the documentation that needs to be collected in order to make the final change proposal derivates from the customer requirement.
  - Change applicability assignment

Finally, a range of Manufacturer Serial Fleet or versions must be assigned to the final change proposal. This may apply only to the customer that originated the change, or a decision can be made to implement the change to other customers/fleets or to products identified by their Manufacturer Serial Number (MSN).

### 3.2 Design change proposals

The design office may propose changes to the configuration for several reasons. Depending on their origin, changes can be justified in three groups:

- Mandatory changes (safety): This type of change is unavoidable because the safety of the product is involved. The consequences of not implementing these changes may lead to catastrophic situations in the operation of the product.
- Improvement changes: Change proposal that improves the functionality or supportability of the product.
- Version definition changes. A change introduced to create new customer versions, in accordance with the customer contractual requirements.

In order to maintain the traceability of changes, it is necessary to consider certain aspects related to these changes and the subsequent actions that are required, such as:

- Change applicability. Provides the applicability of the change in terms of the product.
- Design configuration concepts. In general terms the design office uses the concept of configuration item for a functional item. A functional item is a requirement to comply with a specific function of the product. Since there is not just one way of doing this, it is possible to have different Design Solutions (DS) for the same function. Therefore, there are several DSs associated to one configuration item. A DS can be equipment, a set of equipment, or an assembly of different items that together provide functionality.

The design changes:

- Create configuration items
- Modify configuration items, creating, modifying or deleting new DSs associated to the configuration item
- Drawing and mock-up changes. In some cases, the change does not introduce a new CI/DS, but changes the position of the items (mock-up changes). These changes may also affect the support items (such as maintenance task).
- Changes to design specification and systems descriptions. The functional description of the Product is included in design specifications. Therefore, changes to the configuration of the Product imply change to its functionality, so the design specifications are affected by the design changes. Because design specifications are used in LSA for systems, LSA is impacted by these changes.
Design change status. In order to authorize/implement a design change, an approval process is to be followed. It is important from the LSA point of view to know this status and to act accordingly. As LSA is an integral part of the entire process, the global status of the change must take into account the LSA status. Therefore, the LSA provides feedback to the program change management.

3.3 **Manufacturing deviation from design (concessions)**

Where current approved configuration documentation is correct and manufacturing errors occur, a form for "Deviations and waivers" is completed by manufacturing, either to enable the item to be accepted, or for some kind of repair to be made in order to make the item conform to the drawing or standards, rather than be rejected. Deviations and waivers should not be used to avoid processes required by the life cycle. If non-applicable documents and evaluations are required, these must be tailored and agreed upon prior to contract award.

- **Requirements for deviation**
  
  Prior to the manufacture of an item, if it is considered necessary to temporarily depart from mandatory requirements of a specification or drawings, for a specific number of units or a specified period of time, the customer may authorize a request from the contractor. Items shall not be delivered incorporating a known departure from documentation unless a request for deviation has been approved.

- **Requirements for waiver**
  
  Supplies or services, which do not conform in all respects to the contractual requirements, shall normally be rejected. An item, which through error during manufacture does not conform to the specified configuration documentation, shall not be delivered to the customer unless a waiver has been processed and granted.

3.4 **Supplier changes**

Any change in the equipment provided by any supplier may affect the configuration of the product. The supplier change may affect the equipment or the support items applicable to the supplier equipment.

3.5 **Inservice changes (service bulletins)**

When a modification to the product is to be incorporated once the mission system is operating normally, it has to be introduced through a Service Bulletin (SB).

A SB is a technical document that describes the modification and how it is to be implemented in the product. It has an associated applicability, and once the SB is approved, it can be treated as a design modification.

The difference between a design modification and a SB is that the contractor does not always have information about the implementation of the SB by the customer. Therefore, the applicability of the SB cannot be considered as a true applicability.

4 **The impact of configuration changes in LSA**

4.1 **Evaluation of the impact of configuration changes**

For proper CM, it is necessary to correctly evaluate all of the impacts that the changes may introduce into ILS. The changes can have an impact on configuration items, on the documentation, or on the support system. In order to control the impact of changes on operating and support issues, the following will be taken into account:

- The effect of the change on the ILS disciplines.
- The consequential effects of the change on support elements, (eg spares, test equipment).
- Interchangeability/compatibility of all post- and pre-change items.
If the change has applicability, this will be indicated in the change documentation and a supplementary evaluation sheet will be attached giving details of the assessment.

The following areas must be analyzed for impact:
- Reliability
- Maintainability
- Testability
- Training
- Software
- System development environment
- Standards, plans and procedures
- Spares
- Life cycle costs
- Support & Test equipment
- Technical data/Technical Publication

4.2 Traceability between the source of changes and consequential LSA changes

The traceability between sources changes and LSA changes is achieved by:
- Change proposal criterion traceability
  The change traceability is guaranteed by using as a criterion (trigger) for Change the same code that comes from the design office, or another source, as a change proposal. If this is not the case, it is necessary to maintain a record to relate the criterion used to authorize the changes to LSA/ILS with the original changes.
- Design configuration items versus LSA configuration Items
  Since the design office may use different codes for design configuration than the ones used in LSA configuration, it is necessary to establish a link between the two ways of identifying an item of the configuration. It is recommended to record the correspondence between design configuration items and LSA configuration items. This helps to identify potential impacts in LSA configuration items due to the modification in design configuration.

5 Processing of configuration changes during the LSA process

5.1 General aspects

Once it is decided that a change proposal has an effect on ILS and becomes a criterion (trigger), it is necessary to initiate consequential activities in order to ensure that:
- The impact in the different disciplines is properly identified.
- The different jobs to implement the necessary changes are identified and assigned to every ILS discipline affected by the criterion (trigger).
- These jobs are planned in time and the necessary resources to implement the change are identified.
- The status of the jobs performed by every ILS discipline affected can be known.
- Feedback to the entire configuration management process at the program level about the status of the change implementation is guaranteed.
- Any change proposal identified as non-affecting to ILS needs to be recorded with attached justification. This is required to ensure that the entire change proposal has been analyzed, which guarantees the traceability of changes.

5.2 Coordinating role of LSA

Because LSA integrates the different ILS disciplines, a coordinator with the following responsibilities is required:
- To analyze changes that can affect the ILS
- To determine/assign disciplines affected by the trigger
- To be the focal point of ILS at the program modification committee
- To follow up the status of the trigger in every discipline
- To obtain and provide the necessary documentation from the originator of the change
- To coordinate the jobs between different disciplines and resolve planning problems

5.3 Task assignment by discipline

The coordinator identifies the different disciplines affected. It is necessary to maintain a record of the affected disciplines and the expected jobs to be performed. This is necessary to ensure that any approved change is reflected in ILS. Also, it is useful to ensure that any change in ILS information has a justification.

Once the coordinator assigns the jobs to the disciplines, every discipline needs to analyze and verify that it is really affected by the trigger.

- In the case that the ILS discipline considers that it is not affected, it must inform the coordinator who records this decision, in order to update the impact and eliminate it from the records.
- When the discipline confirms it is affected, the discipline responsible can plan the jobs to be performed, identifying the forecast date for completely implementing the changes.
- In cases where an ILS discipline is affected but requires input from other ILS disciplines, the requirements will be documented and the coordinator will establish coordination between the interrelated disciplines.

5.4 Task status and control

Para 5.3 was about planning and this paragraph is about control. Planning and Control together implement change management. Correct implementation can be tracked by assigning a status to every task discipline or ILS item.

The different codes to be used must be agreed upon at the beginning of the Program and declared in the LCMP document. It is recommended that a minimum set of status codes be used. Following are examples of status codes:

- Open: when a task is identified but it has not already been required to start.
- Pending: when it is required for the ILS discipline to perform the task.
- Close or finish: When all the jobs associated with the task have been correctly completed.

There are different levels at which the criterion status can be managed:

- Criterion (trigger) level. It is the highest level of control. This level is mandatory for proper configuration management at the Program level. It provides the feedback of change management to the customer. A criterion can only be considered as complete when all disciplines declare it as complete at the discipline level.
- ILS discipline level. This provides visibility of implementation in the ILS disciplines. The depth of management level in the organization is optional.
- ILS item level. This level is optional and provides the visibility at the ILS item level. Examples of ILS Item parameters are:
  - Breakdown Element Identifier (BEI)
  - Task
  - ASD system or chapter
  - Data module for technical publications
  - Scheduled maintenance analysis for a system or structural significant item
  - Description of a system

This level provides visibility of the change implementation of an ILS item. Normally, this is sufficient regarding the traceability of a change. This is indirectly obtained from the ILS status.
6 Implementation of configuration management

6.1 Change compilation and traceability
This paragraph explains how to implement the changes (criterion) and enables the main process to guarantee the traceability of the change. This also identifies the ILS elements being impacted and exactly what has been changed. In addition, it must be possible to determine who has or provides the authority to change any part of the ILS elements.

The criterion can impact on the operational system and/or in the support system. It is essential to have a clear process for identification of the operational system configuration and ILS elements or support system configuration. The following figure illustrates how from the source, changes are compiled by the different ILS disciplines, thereby maintaining the traceability of the change/configuration.

![Diagram of Change in design](image)

**Fig 2** Configuration traceability from design configuration to ILS/LSA configuration

6.2 LSA configuration items identification
It is necessary to clearly identify any item or component of the analyzed technical system for ILS purposes. The first important subject to highlight is that not every configuration item from the design point of view is considered an item relevant for LSA activities (LSA candidate). For example, many structural items are not LSA Candidate Items (CI) because they are not considered for support. The rules for identifying LSA CI are already documented in Chap 3.

6.3 Support system and other ILS configuration items definitions
In a traditional logistic approach, configuration refers only to the operational system. However, ILS and LSA consider that the complete project comprises of the operational system and the support system. So the support system must also be considered in the configuration management.
Support equipment also contains components and itself needs other support equipment to maintain it. Where is the limit to consider the CM? The limit is at the maximum level of support required. This means that the level of support analysis influences the CM. This level of support analysis decides if an item is going to be repaired by the operator or not (because it is sent for repair to the contractor/supplier facilities, or it is considered discard). If there are no requirements to repair an item it is not necessary to know its component breakdown. This is the limit for CM. When more than one customer is involved in the business, the limit is established by the customer who requires support at lowest level.

In the case of the support system, it should be considered as an extension of the operational system. If support equipment is not going to be repaired at the operator’s facilities, there will be no need to consider its components for CM. There are other elements of configuration in the support system to be considered for CM such as the technical publication.

According to their nature, training elements can be considered as belonging to support equipment or to technical documentation, as follows:

- Training aids:
  Where there is equipment or systems, such as a simulator or computer based training, training equipments, these can be considered as support equipment

- Courses and training manuals can be considered as special technical documentation

Consumables can be considered as special spares.

Facilities can be regarded as a special category of support equipment. In this case, facilities and support equipment should be linked in a way that there is some support equipment included in the facilities. From the functional point of view, support equipment does not belong to the facilities, but from the physical point of view, they can be considered as components of the facilities. However, it should be pointed out that the requirement of big facilities is a very special aspect. This aspect has to be considered at a very early stage of any project, because normally the construction of facilities and the required infrastructure is associated with high investment.

6.3.1 Technical documents configuration items
The S1000D states that technical documentation is managed with the help of a Common Source Database (CSDB). These DMs are identified by a Data Module Code (DMC). A prepared set of DMs is a technical manual. This provides a capability to create customized technical manuals.

The DM that contains basic information belongs to the functional configuration of the support system (technical data). The physical configuration of the support system for technical data is the DMC, corresponding to the technical manuals and its components are normal DMs.

6.3.2 Illustrated parts list and illustrated parts catalogue configuration items
The S2000M states that the Illustrated Parts Data (IPD) is managed by using specific key data elements to manage the configuration. These also are specific codes.

- CSN - (Catalogue Sequence Number)
- ISN - (Item Sequence Number)

The configuration here is clearly oriented to illustrate how the spares can be disassembled from the product or from equipment in case of equipments with its own Illustrated Part presentation, eg items that traditionally have its own Component Maintenance Publication.

6.3.3 Support equipment configuration items
The term support equipment covers a wide range of items. This ranges from small tools to complex equipment used for testing purposes or training like a simulator. There is no specific ASD specification which covers the subject of support equipment. However, the identification
possibilities previously described are applicable to support equipment also. S1000D can be used for the creation of technical documentation for support equipment in the same way as for the operational system. S2000M can be used to document the support equipment itself and the configuration of the support equipment, if required.

From the system breakdown point of view it is recommended that the support system is separated from the operational system. In an existing SNS for a project it should be determined, how to address and code support equipment. The following aspects should be considered:

- The relationship of the support equipment to a specific project should be visible
- The system where the support equipment is frequently used should be identified within the code

Support equipment, in this way, can be treated like a LRU because it is independent from the support point of view. It can be repairable at first level and can also be treated as having its own IPD representation to illustrate its components. This is a coherent way to assign configuration items with the way used for the operational system. In any case the structure of the breakdown coding for support equipment must be defined and agreed between contractor and customer at the beginning of a project.

From the physical point of view, support equipment can also be included in the facilities configuration. To do that it is helpful to show the capabilities of the facilities in terms of the support equipment available in each facility. One piece of support equipment can be in more than one facility. Likewise one facility may have multiple items of support equipment. In this way it is useful to identify the required facilities for a task according to the support equipment required to perform it. Indirectly it provides the facilities where it is possible to conduct the task.

6.4 LSA compilation by change

6.4.1 General

Compilation by change means that any data compiled has to be done in accordance with a change code that authorizes it and also that change is considered as the driver for the compilation. This is the reason to call the change used to compile criterion (criterion or reason used to change information). There are some basic reasons to compile by change:

- To ensure the traceability of changes produced in design, manufacturing engineering, supplier, or change requirements from the customer. This identifies the real impact of the approved changes.
- To have the authority of any piece of support system changed. This allows the justification of any change.
- To identify the applicability of the items (to be discussed if the part dedicated to an applicability).

The basic way to do it is to identify the impact of any criterion in any item of the operational and/or support systems. Identification of the item affected by the criterion and the type of affectation is recorded for quality assurance purposes. The type of affectation can be:

<table>
<thead>
<tr>
<th>Type of affectation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New item</td>
<td>The criterion creates a new item or extends the applicability of one existing item. The item is new for the range of operational system products indicated by the applicability of the criterion. The range of operational system products is indicated by a Manufacturer Serial Number (MSN) range. This information is very important for achieving final applicability of the Items.</td>
</tr>
<tr>
<td>Type of affectation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Deleted item</td>
<td>Most of the time it is not a real deletion, it is only a way to say that the existing item is not any more applicable to the products covered in the range indicated by the criterion applicability. This information is very important for getting the configuration item applicability.</td>
</tr>
<tr>
<td>Modified item</td>
<td>Used when only traceability function is required for the change, and non-applicability impact is declared. For example a task is changed by a criterion but the task is still applicable to the same range of MSN that was previously to the application of the criterion. What is affected and has different applicability are the components of the task (spares, step description, etc)</td>
</tr>
</tbody>
</table>

The following Fig 3 illustrates the explanation above.

**Fig 3  LSA compilation by change**

### 6.4.2 Operational system configuration compilation

The LSA operational system configuration is linked to the design configuration of the operational system.

- Through approved design modifications that have impact in ILS/LSA and for that reason they become criterion. This modification must be recorded and associated to the breakdown revision identifier, specifying whether to create a new breakdown element, deleting existing ones (eg in the way of restricting applicability of the breakdown element) or just extending the range of applicability.
- By linking the breakdown revision identifier, the hardware element realization and product variant realization applicability and the design configuration items, there are different ways to identify a configuration item in design. The following are examples of how to identify an item.

- **Configuration item number**
  
  It is a number to identify a function or assembly position. It is equivalent to the BEI.

- **Design solution**
  
  Corresponds to the different alternatives of design that can be analyzed or mounted for different MSN. It is equivalent to the hardware element realization. In some cases
these do not correspond to a single equipment and it is an assembly of items. When this happens it is necessary to have extra information, like the next element of identification.

- **Parent-child relation**
  Identify one single position in the assembly. The parent corresponds to the Part Number (P/N) of the assembly and the child corresponds to the P/N of the component. In some cases like mock-up applications the child position corresponds to a single element. In other words if one P/N is mounted twice in an assembly there are two parent child relations with the same P/N child. In these cases the breakdown element structure is used to relate one or more parent-child relations but having the same P/N child.

Maintaining the relation between the breakdown element revision, hardware element realization and product variant realization applicability and the original design configuration identification provides the means to guarantee the traceability of the LSA configuration. It also provides help to identify BEIs potential affected by new design changes.

### 6.4.3 Functions and descriptions configuration

Functions and descriptions of systems, subsystems and equipments are subject to changes resulting from design, manufacturing or suppliers changes. To translate this change to the LSA description and functions requires maintaining a traceability of the changes. So the criterion is also used to determine the impact upon the element used to describe or define the functions of the operational system items.

- Descriptions are basically a set of paragraphs. So to configure a description it is necessary to have a paragraph identifier. It is recommended to manage configuration only at first level of paragraph and not to subparagraph level.
- Functions also need to have identifiers. It is possible to have functions and sub functions. The complexity of managing both function and sub function is excessive, therefore it is recommended to manage only the configuration of function.

The way to manage the configuration is the same as Breakdown Element Revision (BER), and product variant realization applicability. However, in place of BER, a paragraph or function identifier is used and in place of product variant realization applicability, paragraph variant or function variant must be used.

In any case the depth of CM (function, sub function) must be decided at the beginning of the project.

### 6.4.4 Failure modes and effects analysis configuration

The failures analysis is affected by the configuration changes, so it is necessary to know how the failure analysis is affected by the criterion for change. There are three parts of the failure analysis that can be affected, and need to have configuration management:

- **Failures configuration**
  Refers to the change in the failure modes that can be changed due, primarily to the design changes. These changes may occur due to:
  - Different equipment can produce different failures
  - Different equipment can produce the same failure but with different failure rate
  - Different equipment provides different Built In Test Equipment (BITE) capabilities or different detectability

  So it is necessary to identify the failures and also variant of failures. It is important as well to assign the criterion, which authorize the new failure or variant, and the criterion that make non-effective an existing failure.
− Causes configuration

The cause of a failure is an LRU/equipment when the analysis is performed at the system level or one SRU/component when the analysis is done at equipment level. As the causes are items of the operational system configuration, the cause is affected by the same changes that influence the operational system configuration. The causes are identified by BER and hardware element realization. In this case the criterion does not need to be associated to the cause because it can be obtained indirectly from the affected BER and hardware element realization.

− Warning and error messages configuration

Related to the capabilities of BITE appears some information that is provided by the central maintenance system (error messages) and the indicating system (warnings presented in the warning display). These messages can be different from one operational system product to another. These differences are due to the design changes or supplier change introduced to the equipment with BITE. Criterion must be associated to the new or modified messages/warnings like in the other cases.

6.4.5 Task analysis configuration

Task analysis is also subject to CM. Different products imply different applicable tasks. A criterion can create a new task, restrict the applicability of the task or modify its components. Components of a task are considered the following items:

− Spares

Spares are items of the operational system configuration, so they can have a BEI or a hardware element. Changes in design can affect the applicability of the spares thereby effecting the identification of the spares used in a task. For this reason, it is recommended that a criterion for spares be established.

− Consumables

These are not necessarily components of the operational system because they are consumed during the task. In other cases the item used in the task remains as part of the operational system. In this case it is recommended that the criterion used to change the supplies or consumables used in the task are identified (the term supplies is used in S1000D while the term consumable is used in S2000M).

− Support equipment

Due to changes in the configuration, the required support equipment can be different even when the configuration item where the task is applicable has not changed. In this case, what can occur is that the environment around the item has changed, so that a different tool will be needed. Therefore, criterion must be associated to the changed support equipment applicable to the task.

− Zones and access panels

One configuration item can be located in a different area zone of the operational system or just the area around it has changed. In these cases the task (e.g., replacement of a LRU) can require different panels to be opened or the task is performed in a different zone of the aircraft. So it is necessary to associate the criterion to the access panel’s changes or zone changes.

− Personnel configuration
Required persons can be different because there are changes in other components of the task or the time required to perform the task is different. In this cases the changes included in the task need to be authorized by a criterion.

- **Subtasks configuration**

  In the subtasks description the detailed instruction of how to perform the task is given. It is recommended that the configuration is managed at the level of subtasks, otherwise it becomes unclear how to manage the changes. When a subtask within a maintenance procedure needs to be changed a new variant of the subtask and/or task needs to be created. Then it is important to have a variant identifier when a criterion substitutes one existing subtask by another one. The criterion identifies the existing one as "out" or "old" and the new one as "new" or "in".

### 6.4.6 Scheduled Maintenance analysis configuration management

The Scheduled Maintenance Analysis (SMA) process is also affected by the CM, in the analysis and also in the results of it. SMA uses the failure modes as source for the analysis and the result of it is the scheduled maintenance task.

- **SMA logic configuration**

  The logic is oriented to the failures mode/effects. Since failures modes are configured, the logic of the SMA is also configured and traceability of this analysis is required identifying the effect of criterion implementation.

- **SMA disposition configuration**

  Disposition is linked to the causes, so different causes introduced by a criterion can derivate in a different disposition.

- **Other SMA configuration aspects**

  The results of the SMA are the scheduled tasks. These tasks can be different as a consequence of a change (criterion).

### 6.4.7 Support equipment change compilation

Support equipment is part of the support system. They belong to the configuration of the support system but also they have configuration because they have components.

- **Support system**

  The support equipment belonging to the support system does not have direct applicability. The applicability is derived from the task where they are used. Support system has function and the support equipment covers these functions. In the same way as equipment for operational system, there can be more than one alternative to comply with the support system function. This means that several pieces of equipment can have the same BER and different hardware element realizations. Support equipment will also have design evolution and therefore product variants of one support equipment can be found.

- **Configuration inside support equipment**

  There are modifications from a supplier of support equipment that authorize changes to the support system. Support equipment configuration is managed in the same way as the other items of the operational system. So the recommendation for this support equipment is to treated as operational system equipment configuration. In these cases product variant is recommended. In some cases the support equipment could be a very large piece of equipment and the applicability of their components needs to be oriented to MSN of the equipment. In other words the support equipment is like a product.
6.5 Applicability management

6.5.1 General

It has been defined how to manage the change and how to create and maintain configuration items for the whole system (operational and support systems). The configuration is not unique and every product can have different components. Therefore it is required to know the configuration of every product or, the other way around, for every item of the configuration the products where it is assembled. This is provided by means of an applicability management.

The following figure shows how to obtain the applicability of a configuration item (in this case a DM is used as an example) using the applicability of the criteria assigned to a configuration item. Nevertheless, DM is S1000D terminology and is used here for any package of ILS/LSA package of information. Examples of this DM usage are systems descriptions, maintenance tasks, Failure Modes and Effects Analysis and reliability data.

This scheme will be explained in detail in the following paragraphs. Basically when a set of criteria is applied to a DM the applicability of every criterion in this set is transferred to the DM creating the applicability or applicability of the DM and also the applicability of the components (in the case of a maintenance task: Spares, support equipment, steps, personnel, etc).

6.5.2 Manufactured serial number and fleet serial number concepts

From the customer point of view, operation or service are the targets for a system. In the end, the operation is complied by a single product. A Manufactured Serial Number (MSN) identifies this single product. It is a unique number to identify the product. It is mandatory that every product has an MSN. Therefore, the range of MSN where it is mounted mainly identifies the applicability of an item.
The Fleet Serial Number (FSN) is another possibility. The concept of fleet is based in the combination of two concepts: customer and version. Version is a set of products with basically the same configuration because they are thought to comply with the same type of missions or with the same operational characteristics. Minor differences can occur between the products belonging to a version due mainly to evolutions or improvements. The conjunction of a product for a customer is defined as the fleet. A product is identified by customer + version + serial number inside version.

**Fig 5  MSN usage example**

There must be a correspondence between MSN and FSN.

**Fig 6  Fleet example**

There must be a correspondence between MSN and FSN.
S2000M defines the applicability by fleet numbering, providing the service (code to identify the customer) version (e.g., single seat aircraft, twin seat aircraft) and serial number inside the version.

Due to the planning for manufacturing, normally the numbering for MSN and FSN are not coincident. The first MSN is for one customer then the second MSN is for other customer and so on.

6.5.3 Effectivity of change/criterion

A change proposal can be applicable to all products or can be limited to one or more range of products. This range is defined by two figures:

- **MSN from**
  - Identify the first MSN where the change is applicable. This item is included.

- **MSN to**
  - Identify the last MSN where the change is applicable. This item is also considered included.

Other possibility of expressing the applicability of change/criterion is providing the version and then the range inside the version. It is possible that a change/criterion apply to more than one version/range.

6.5.3.1 Example: Design criteria applicability to LSA applicability

In the traceability management process, it has been described how to indicate the affectation a criterion has to a configured item. The following example shows how to convert this into applicability ranges for an item.

- **First step:** The first criterion called criterion 1 is defined having the applicability by MSN from 0001 to 9999.
- It is used to create a new BER of A212101 and for a new hardware element realization of P/N1. Criterion 1 makes applicable ("in") the range of serial numbers.
- For that affectation BER of A212101 has an applicability from 0001 to 9999.
- A new criterion with an applicability 0020-9999 creates a variant for the existing BER with a new hardware element realization of P/N2 and the same product variant with a new block of serialized items.
- As a result of it:
  - breakdown element revision of A212101 and hardware element realization of P/N1 is only applicable from 0001 to 0019
  - breakdown element revision of A212101 and hardware element realization of P/N2 is only applicable from 0020 to 9999
6.5.3.2 Example: Design criteria applicability to IP/IPD configuration item (CSN/ISN) applicability

The following figure shows an example of applicability calculation for IP/IPD items based on CSN and ISN.

![Applicability Calculation Diagram]

**Fig 8 Applicability calculation**

### 6.5.4 MSN versus FSN applicability management

MSN or FSN are used when the product or element with a change in its configuration does not change its P/N. In this case the MSN or FSN is used to distinguish between the different configured end items. For example, where the P/N of the aircraft is not changed, the MSN or FSN is required to identify the different aircraft manufactured. In the case of an equipment where the configuration of the equipment is changed, the P/N of the equipment is also changed. Then the UOC is used to identify the applicability of the component of every equipment identified by different P/N.

MSN applicability is oriented to the manufacturing. It is a number that does not change during its life. Normally the applicability in design is given by this way so criteria coming from design will usually have this type of applicability.

For FSN applicability, it is recommended that the rules and definition of S2000M are used. It is recommended that the use this type of applicability is utilized because it is easy to customize the information that is to be delivered.

### 6.5.5 Product variant management

Product variant (formerly represented by a Usable on Code, UOC) is used mainly for equipment or other items where the configuration is not managed by MSN. In this case, when changes occur, the P/N of the item also changes. This is a method to identify the configuration of a P/N configured item similar to a color code (mainly used for equipments configuration management).

Product variant is used to identify each model of a product. The hardware element revision can be associated with each applicable product variant to form the overall breakdown of a product. **Fig 9** is an example of using an applicability code (e.g., like a UOC) for the product variant.
**6.5.6 Serialized item management**

Serialized Item Management is a requirement for delivery configuration management. This is accomplished by marking populations of select items (parts, components, and products) with a serial number identifier, enabling collection and analysis of maintenance data about the items. As a minimum, it is appropriate to consider selecting item populations from within the following categories:

- Repairable items down to and including sub-component repairable unit level
- Life-limited, time-controlled, or other items with records (e.g. logbooks, aeronautical equipment service records)
- Items that require technical directive tracking at the part number level

For these serialized items it is necessary to provide the delivery configuration. This delivery configuration is a list containing the following information:

- MSN of the product
- BER of serialized item
- Part number of serialized item
- MSN of the serialized item

**6.6 Production and releases management**

**6.6.1 General**

The main objective of the CM is to know the build standard of the product (operational system and support system). It is necessary to control the changes of a component of the end product as these changes may, in time, result in changes to the end product itself.

This includes a set of elements to be managed, like:
− End product identification
− Management of end product releases
− Change traceability of the releases
− Customization of releases

6.6.2 End product identification
It is necessary to manage the identification of the different end products to be delivered to the customer. In the end product identification two factors need to be considered:

− Function or operation of end product that implies the use of end product type identification
− Timing evolution due to change implementation

6.6.2.1 End product type
Every end product has two components in its identification depending on the type of differentiation used to make the classification:

− Operational system
  It is the system that performs the mission required by the customer (e.g. passenger aircraft, or military transport aircraft or combat aircraft). The model and the version are used to identify the different types of operational system products.

− Support system
  It is a system used to support the operational system. As the type of support to be provided is different, a more detailed classification is needed. For example, technical manuals have different functions. They serve as an operational manual used to provide the required information to operate the operational system and as maintenance manuals used to repair the operational system. Each must be managed in harmony with the operational system. Similarly, the LSA database can be considered as a kind of support system product storing the results of the LSA activities.

6.6.2.2 Versions, releases and issues of a product
Depending on the changes included as a consequence of time evolution of the end product due to design changes or customer requirements changes, another type of identification must be included. This is the version identification. So the end product is identified by means of the type of product plus the version or release identification.

− Concerning operational system
  This is identified by the version or more exactly by the MSN. When several MSN have the same configuration they can be grouped in a version. But the version does not ensure that all the Products included in the version have exactly the same configuration, as it was explained previously.

− Concerning support or training equipment
  The product is identified by a P/N or for complex equipment it is also identified by a MSN.

− Concerning support information (e.g. LSA database, technical documentation, training courses) version, issues or release numbers are used to identify the product.

An identification process is required for the management of the different releases or versions (there are other possibilities of defining versions depending on the function of the item such as single seat combat aircraft or double seat combat aircraft for pilot training purposes). Rules for proper releases identification must be included in the CM plan document.

6.6.3 Change traceability of releases
As it has been stated, the concept of the release is developed because there are changes occurring over time. Due to this, it is necessary to know what is included in every release of a
product. In other words, a list of criterion must be provided for every release. This list provides the traceability of changes included in the release, as well as, the authority for this release.

Normally, it is required to identify any component of the release affected by a criterion. For example an aircraft maintenance manual is an assembly of DMs. This requirement implies that, when a new release is issued, it is necessary to identify the DMs that have been modified regarding previous issues, as well as the criterion or criterions that have led to that change.

6.6.4 Release management
Release management ensures a proper CM of the releases. The following is a list of the requirements needed to ensure efficient management and quality of a release.

− A release coordinator is required to organize the tasks to be performed in order to provide the customer with a coordinated issue of the new release of all end products that are delivered at a milestone. This job is linked to the high-level CM activities.
− A set of tasks must be included covering the activities to be executed by the different ILS disciplines in order to produce the support end products which are under their responsibility. This end product must be prepared in a coherent way (eg the aircraft maintenance manual must be in accordance with the IPD. It would not be acceptable that the aircraft maintenance manual makes reference to a spare that is not in the IPD. The criterion used would be that spares in a task of the aircraft maintenance manual must be in the IPD.

Because the release is defined as the changes included from the last delivery as a consequence of implementing a set of criterion, a mechanism must be used to link the release management to the criterion management. Only when the criterion included in the release has been fully applied, the release can be considered as ready for delivery. This job is linked to the detailed CM.

− The status of the tasks provides a visibility of the status of a release.

The following figure illustrates the release management as the management of a set of activities to be performed:

− Tasks oriented to the preparation of the deliverables. Every one of these tasks is another set of tasks oriented to the jobs to be done to prepare the release.
− Tasks oriented to the criteria. There is one task per criterion used to define the new release. So one release can be understood as a set of criteria. This process links the release management to the change management.
6.6.5 Customization of a release

When a release is delivered to a customer it is clear that the release is unique for that customer. So the contents of every release are not only influenced by the criterion included in the release elaboration but depending on the customer, the contents of the release are different.

The applicability of the end product components is the key to provide customization. Since the components have applicability referred to the mission product identification (MSN or versions or customer fleet), it is possible to discard any component that it is not applicable to this customer. Criterion that does not affect a customer should not be included in the release definition.
Chapter 5

Influence on design

Table of contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Influence on design</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>2 Design considerations</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Availability</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Reliability</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Maintainability</td>
<td>5</td>
</tr>
<tr>
<td>2.4 Testability</td>
<td>5</td>
</tr>
<tr>
<td>2.5 Prognostics</td>
<td>5</td>
</tr>
<tr>
<td>2.6 Standardization</td>
<td>6</td>
</tr>
<tr>
<td>2.7 Interchangeability</td>
<td>6</td>
</tr>
<tr>
<td>2.8 Environmental considerations</td>
<td>6</td>
</tr>
<tr>
<td>2.9 Human factors/ergonomics</td>
<td>7</td>
</tr>
<tr>
<td>2.10 Obsolescence</td>
<td>7</td>
</tr>
<tr>
<td>2.11 Supportability</td>
<td>7</td>
</tr>
<tr>
<td>2.12 Cost effectiveness</td>
<td>7</td>
</tr>
<tr>
<td>2.13 Software design</td>
<td>8</td>
</tr>
<tr>
<td>3 Development programs/projects</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Strategy for LSA influence on design</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Design influence on development programs</td>
<td>8</td>
</tr>
<tr>
<td>3.3 Supplier design</td>
<td>9</td>
</tr>
<tr>
<td>3.4 Critical/milestone design reviews</td>
<td>9</td>
</tr>
<tr>
<td>4 Checklists</td>
<td>10</td>
</tr>
<tr>
<td>4.1 Selection of parts/equipment</td>
<td>10</td>
</tr>
<tr>
<td>4.2 Reliability</td>
<td>10</td>
</tr>
<tr>
<td>4.3 Maintainability</td>
<td>11</td>
</tr>
<tr>
<td>4.4 Testability</td>
<td>11</td>
</tr>
<tr>
<td>4.5 Prognostics</td>
<td>11</td>
</tr>
<tr>
<td>4.6 Standardization</td>
<td>11</td>
</tr>
<tr>
<td>4.7 Interchangeability</td>
<td>11</td>
</tr>
<tr>
<td>4.8 Environment</td>
<td>11</td>
</tr>
<tr>
<td>4.9 Human factors/ergonomics/accessibility</td>
<td>11</td>
</tr>
<tr>
<td>4.10 Obsolescence</td>
<td>12</td>
</tr>
</tbody>
</table>

List of tables

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>References</td>
</tr>
</tbody>
</table>

List of figures

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Opportunity to influence design and support cost during product lifetime</td>
</tr>
<tr>
<td>2</td>
<td>Breakdown of availability as an approach to structure design influence</td>
</tr>
<tr>
<td>3</td>
<td>Effectiveness - balancing availability and life cycle cost</td>
</tr>
</tbody>
</table>

Applicable to: All

S3000L-A-05-00-0000-00A-040A-A

Chap 5

DMC-S3000L-A-05-00-0000-00A-040A-A_001_00_EN-US.doc

2010-04-01 Page 1
1 General

1.1 Introduction

This chapter covers the subject of LSA parameters influencing the design of the Product.

It should be a principle for the staff involved in the LSA and in the related ILS disciplines to claim the influence on design as one of the main goals of supportability engineering. For staff involved in design (software or hardware), Influence on Design is important in order to interact between the Product design program and LSA program.

1.2 Objective

The broad objectives of LSA are to influence design, create the most effective support concept, and define logistic support resource requirements. In this chapter, the focus is put primarily on the influence of the Product design; however, it is not possible to isolate it from the iterative work needed for an effective support solution or from requirements on logistic support resources.

General objectives for the Product must be translated into more specific requirements for an individual project. The key to a productive and cost effective LSA is the concentration of available resources on activities which most benefit the program. Such concentration might be called the analysis strategy. When deciding on an analysis strategy, it is necessary to consider the type and scope of the development project. The possibilities of influence may also vary due to previous design decisions and the life cycle status.

The opportunity for influencing design in order to fulfil LSA requirements is at its highest in the beginning of a project, during the conceptual phases. Early design influence can reduce or eliminate the need for later design changes (need for redesign) in order to make the Product fit for operation and support. LSA requirements are considered useful for designing a new product with regards to total system availability and cost effectiveness. The purpose is to influence the design with LSA requirements in a manner similar to how the design of the support system is influenced by the primary product design and requirements. This is done in a structured way within a project through reviews and baselines.

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 7</td>
<td>LSA Failure modes and effects analysis</td>
</tr>
<tr>
<td>Chap 8</td>
<td>Damage and special event analysis</td>
</tr>
<tr>
<td>Chap 13</td>
<td>Software support analysis</td>
</tr>
<tr>
<td>Chap 17</td>
<td>Disposal</td>
</tr>
</tbody>
</table>
## 1.3 Scope

The scope of this chapter is to describe:

- design parameters to be influenced by LSA and vice versa
- how to make a strategy for LSA influencing design
- the perspectives of supplier and vendors
- reviews of milestones, critical or not
- the benefits of LSA influencing the design

The approach to influence the design with the parameters of availability is applicable on systems and subsystems within the Product as well as the support systems design and development.

## 2 Design considerations

Many considerations could be made upon design; the following are a list of features which could be used to influence the design of a primary product with regards to making the end product/system more efficient and cost effective.

### 2.1 Availability

Availability is the measure of the degree to which an item is in an operable and ready-for-use state at the start of a mission or operation, when the mission or operation is called for at an unknown time. This is sometimes called operational readiness.

Reliability, maintainability and supportability all contributes to the availability of a system.
The inherent availability of a system is driven by the reliability and maintainability of the Product. It could be described as the probability that a system, when used under stated conditions in an ideal support environment (ie no lack of support resources) will operate sufficiently at any point in time. It excludes preventive maintenance, delay times and is expressed as

\[ A_i = \frac{MTBF}{MTBF + MTTR} \]

where MTBF is the mean time between failure and MTTR is the mean time to repair.

The achieved availability is similar to inherent availability with the exception that preventive maintenance is included. It is expresses as

\[ A_a = \frac{MTBM}{MTBM + M} \]

where MTBM is the mean time between maintenance and \( M \) is the mean active maintenance action time.

Operational availability is achieved through the combination of the Product and its support system. It could be described as the probability that a system, when used under stated conditions in an actual support environment, will operate sufficiently when called for. It is expresses as

\[ A_o = \frac{MTBM}{MTBM + MDT} \]

where MDT is the mean maintenance down time.\(^1\)

Here is a most important point - the availability of the Product, is effectively gained by putting requirements on both the Product and the support system, and is achieved by the combination of the two through maintenance actions.

---

2.2 Reliability
Reliability is a prime driver of support resources. It relates to the duration or probability of failure-free performance of a Product under stated conditions, or the probability that an item can perform its intended function for a specified interval under stated conditions.

Products with high reliability are usually cost effective. The design of a Product could be made that reduces and compensates for the implication of a low reliability so when a failure occurs, it is easy to locate the failure and easy to perform corrective actions. To maintain availability, it is to some extent possible to compensate low reliability with increased maintainability and supportability.

2.3 Maintainability
Maintainability is the measure of the ability of an item to be retained in or restored to a specified condition, when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance and repair.

Characteristics of maintainability are, for example, the time it takes to replace or repair a Product or a system in order to restore or maintain its functionality, the amount and complexity of support resources that are required and the required skill levels of the personnel performing the activities.

The design could be sensible or be able to withstand the environment in different degrees. For example the design could be specific to withstand an operational and maintenance environment in airports/airbases, loading/unloading with support equipment or to withstand sandstorms, hail, and salt laden environments. For possible inputs to consider, please refer to Chap 8.

For maintainability aspects for software, such as software loading, please refer to Chap 13.

2.4 Testability
Testability is a design characteristic which allows the status (operable, inoperable, or degraded) of an item and the location of any faults/failures within the item to be confidently determined in a timely fashion.

The design solution to fulfil testability requirements could be achieved by a test system integrated into the Product; or the test system as a part of the support resources requiring an interface to the Product in order to locate failures.

Carefully designed test systems can reduce the work needed to locate and correct failures/faults and can also be used to verify full system functionality.

Redesign driven by testability requirements, from the LSA FMEA, is a highly complex activity. Milestones in the design program should be added to allow for necessary iterations. For more information on LSA FMEA please refer to Chap 7.

2.5 Prognostics
Product parameters and data could be used in models to predict maintenance need or repair. The functionality of maintenance or repair prognosis could either be part of the Product or the support systems each have different advantages.

The benefits of prognostics are increased means of knowing a Product “health” status and, based on that knowledge, avoid critical failures and plan for activities to maintain system functionality.

In a system that does not have a means of prognostics; preventive maintenance is used to prolong the system lifetime and system safety, and is balanced against the corrective maintenance actions and resources.
2.6 Standardization

It is usually cost effective to use standard equipment instead of equipment which is specially designed. Development/design cost and time is reduced when using an existing part or system that fulfils the requirements.

Utilization of existing logistic support resources, which fulfil the requirements, could also be possible when using standardized equipment and reduce the need for investments in design and acquisition of support systems resources. Design decisions that lead to the need of specially developed SE should be considered carefully since such equipment drives cost.

Other positive effects related to this could be increased product/system maturity and knowledge and an increase of mobility of the operational unit.

Factors that support the potential benefits are the following:

- Use of existing items avoids the development costs that would be incurred to develop new support resources
- Cost to develop new training programs may be avoided
- Commonality of support resources may increase the availability on support resources and reduce the logistics footprint
- Use of standardized items reduces the development time required to determine support resource requirements
- Personnel proficiency in using support and test equipment can be increased through an increase in frequency of use of the same item, rather than having to learn how to use different items

Standardization includes the requirement of interchangeability.

Software design could also be given standardization requirements and could be put on languages, information structures or software and information carrying media for example.

To achieve the benefits, requirements on standardization must be established prior to initiation of the design effort so that the cost of designing or redesigning to meet requirements can be minimized.

2.7 Interchangeability

Products should be designed so that the equipment, components and parts are interchangeable within or between Products, where applicable. The purpose is to increase flexibility of usage of the equipment and a reduction of spare stock levels.

To achieve the benefits, requirements on interchangeability must be established prior to initiation of the design effort so that the cost of designing or redesigning to meet requirements can be minimized.

2.8 Environmental considerations

Known operational environment and maintenance environment information could influence the choices of equipment and design solutions. Temperatures, humidity, sandy and salty environments, are examples of parameters that influence operation and maintenance.

It is also of interest to ensure that the Product fulfils requirements in-between operation- during transportation, assembly and disassembly and storage. Taking these considerations into account increases the robustness.

Chemicals/materials needed for maintenance, operation and disposal from an environmental perspective are also important. The implications to include hazardous materials, hazardous waste, and environmental pollutants could have an impact on the Life Cycle Cost (LCC) as well as the safety of personnel in contact with the Product. For more information and depth on this subject, please refer to Chap 17.
2.9 **Human factors/ergonomics**

Requirements on the Product for human factors and ergonomics could be formulated. The Product could be designed in such a way that the accessibility for operators and technicians are taken into consideration. Activities during operation and maintenance with humans in the loop should be given consideration. Handling qualities, for example, parameters such as weight and size could be relevant drivers for support resources in order to physically be able to perform activities.

Design tools can be used to analyze (virtually or by other means) the accessibility for human and the disassembly and assembly activities.

Chemicals/materials as part of the Product or needed for maintenance, operation and disposal are also important from a human factors perspective. The implications to include hazardous materials and environmental pollutants could have an impact on the safety of the personnel and the need for support resources.

Product interface requirements such as access panels for technicians or maintenance personnel and operator stations requirements are a joint interest for LSA and design.

2.10 **Obsolescence**

Depending on the Product life cycle length, it could be useful to consider the possible situation of obsolescence during the design phase. Where obsolescence occurs, actions will be necessary to keep the functionality. Examples of typical actions are re-design/modification, lifetime purchasing activities or scrapping of the system and replacement with a new system. When designing or choosing systems or sub-systems to a design, it is useful to keep in mind the future replacement of an item due to obsolescence.

2.11 **Supportability**

Supportability is the measure of the degree to which all resources required to operate and maintain the Product are provided in sufficient quantity. Supportability encompasses all elements of ILS support resources such as technical information, support equipment, spares, personnel etc.

Through careful system design, the amount of resources required for supporting a Product can be reduced by considering maintainability and reliability. Product downtime, in reality consisting of active maintenance time, logistics delay times and administrative delay times can be reduced. A Product that requires a lot of support resources is more exposed to the risk of shortage of support resources, waiting or queuing for resources.

2.12 **Cost effectiveness**

The design and the choices involved in design have a cost implication for the life cycle cost. Life cycle cost could be broken down into design/development, production/procurement, operational and disposal costs. Comparative analyses between different alternatives of design solutions, including the support resources, are required. The characteristics mentioned previously in this paragraph and cost are necessary to balance availability and LCC.
2.13 **Software design**

Software is usually a part of modern Product design. Certain considerations could be made to influence the software design in order to give the software some of the features mentioned previously in this paragraph.

Considering the consequences of software support actions described in Chap 13, software support analysis, decisions can be made regarding how the software should be designed and implemented in order to achieve the availability requirement of the system.

3 **Development programs/projects**

Development programs/projects should include LSA and all efforts should be done in a documented, structured, interactive closed loop process.

A representative for LSA should be appointed and part of the project team.

3.1 **Strategy for LSA influence on design**

The key to a productive and cost effective analysis effort is the concentration of available resources on activities which most benefit the program. Such concentration might be called the analysis strategy.

The broad objectives of LSA are to influence design, structure the most effective support concept, and to define logistic support resource requirements. These general objectives must be translated into more specific objectives for individual projects. The amount of design freedom (flexibility) and opportunity of design influence varies with the type of program/project and is the main input when deciding on a strategy.

Objectives and the analysis strategy should be iterated, refined and balanced against available resources until they become firm program goals or requirements.

3.2 **Design influence on development programs**

The very nature of development programs may vary from

- new development programs where the Product and support products are developed from the very beginning

or

- be system or subsystem design changes/improvements of an existing Product

or
– be a fast track program using existing technology developed in-house or by a supplier.

The amount of design freedom and possible design influence therefore shifts. Often, a development program of a complex Product is a combination of the different types. LSA effort and objectives should be focused accordingly.

Furthermore, the possibility of design freedom may exist for the support system but not the Product, and vice versa. The LSA objective of making reliability, maintainability and supportability requirements an integrated part of Product requirements and design can best be achieved if designers are oriented toward reliability, maintainability and supportability objectives commencing with the design effort.

3.3 Supplier design

Supplier design, design performed by a vendor or subcontractor, should be approached in the same manner as in-house design.

It is important to provide the supplier with LSA goals and requirements specific to the supplier in order to influence supplier design before the design efforts commence.

In conjunction with design influence by requirements and goals, the requiring party must initially decide and specify the LSA tasks that are to be done by the supplier, which tasks are to be shared between the requiring party and the supplier, and those that are to be performed solely by requiring party. Once done, the LSA portion of the contracting plan can be developed and work requirements written into the procurement documentation.

It is very useful to allow the prospective performing activities, under the bidding terms of the procurement, to recommend adding or deleting LSA tasks and to provide a more detailed subtask definition and schedule.

Additionally, prospective performing activities should be encouraged to make use of cost effective data generation procedures. Acquisition program objectives must be considered in preparing procurement documents. For example, in technology demonstration procurement, one may specifically exclude certain LSA task requirements. Supportability objectives for this type of procurement would best be served through design influence and generation of an LSA data for subsequent detailed analysis efforts when the technology is utilized. If the acquisition program is oriented to develop and procure a Product, then other LSA tasks become equally important.

3.4 Critical/milestone design reviews

It is of great importance to establish and document design review procedures (where procedures do not already exist) which provide for official review and control of released design information with LSA program participation in a timely and controlled manner.

These procedures shall define accept/reject criteria, the method of documenting reviews, the types of design documentation subject to review, and the degree of authority of each reviewing activity.

Program/project planning must coordinate the design and development reviews and the LSA reviews.

Formal review and assessment of LSA requirements shall be an integral part of each Product design review. Results of each Product design review shall be documented. Design reviews shall identify and discuss all pertinent aspects of the LSA program.

Technical information generated and documented during the design process must be disseminated among designers and supportability specialists in order to surface interface problems between design concepts and operators, maintainers, and support equipment. Technical design information such as diagnostic features, interfaces, reliability estimates,
obsolescence evaluations, and item functions, which determines supportability, should be an integral part of design documentation.

Scheduled reviews should be performed "bottom up" to include subcontractors and suppliers, as appropriate.

Agendas shall be developed and coordinated to address the relevant topics as they apply to the program phase activity and the review being conducted.

Examples of topics are:

− LSA conducted by task and WBS element
− LSA assessment of proposed design features including supportability, cost, and readiness drivers and new or critical support resource requirements
− Corrective actions considered, proposed, or taken, such as:
  • Support alternatives under consideration
  • System/equipment alternatives under consideration
  • Evaluation and trade-off analysis results
  • Comparative analysis with existing Products
  • Design or redesign actions proposed or taken
− Review of LSA requirements - supportability (with review of specifications as developed)
− Progress toward establishing or achieved goals
− LSA documentation required, completed and scheduled
− Design, schedule, or analysis problems affecting LSA

4 Checklists

To facilitate design work and reviews, the following checklists can be used as a starting point and input. However, creativity in defining checklist questions and in adjustments to the program in question is absolutely necessary. Often, the checklists can be defined through dialogue between the LSA and design disciplines.2

4.1 Selection of parts/equipment

− Have appropriate standards been consulted for the selection of parts?
− Have the selected parts/equipment been evaluated with regards to reliability, maintainability and supportability?
− Have supplier sources for component part procurement been established?
− Is the supplier reliable in terms of quality level, ability to deliver on time, etc?

4.2 Reliability

− Has the system/equipment wear-out period been defined?
− Have failure modes and effects been identified?
− Are item failure rates known?
− Have parts with excessive failure rates been identified?
− Has mean life been determined?
− Has equipment design complexity been minimized?
− Is protection against secondary failures (resulting from primary failures) incorporated where possible?
− Have reliability requirements been met?

4.3 Maintainability
- Are the total numbers of different kinds of fasteners minimized?
- Are the fasteners used standard items?
- Have the fasteners been selected based on the requirement for standard tools in comparison with special tools?
- Are equipment identified as replaceable items?
- Is the time it takes to replace an item minimized?
- Is consideration taken upon the operational environment, such as hail storms, sandy and salty environment?
- Is consideration taken on how to load software, loading times and media?

4.4 Testability
- Have self-test provisions been incorporated where appropriate?
- Is the extent of self-test compatible with LORA?
- Is the self test automatic?
- Have direct fault indicators been provided? (eg light, message)
- Are test points/interface provided to enable check-out and fault isolation beyond the level of self test?
- Are test points/interface accessible?
- Are test points/interface functionally or conveniently grouped to facilitate sequential testing?
- Are test points/interface provided for direct test of replaceable items?
- Are test points labelled?
- Can every equipment malfunction be detected by a no-go indication at system level?
- Will software provide adequate test information?

4.5 Prognostics
- Is the functionality to predict maintenance need identified for the equipment?
- Are the parameters used to predict maintenance need identified and obtainable?
- Is the sampling frequency of the parameters used for prognostics adequate?

4.6 Standardization
- Are standard equipment/parts incorporated in the design to the highest extent possible?
- Are the same parts used in similar applications?
- Is the number of different part types used throughout the design minimized?
- Are identifying labels and markings and nomenclature standardized to the highest extent?

4.7 Interchangeability
- Are modules and components that have similar functions electrically, functionally and physically interchangeable?
- Are components with the same part number but provided by different suppliers completely interchangeable?

4.8 Environment
- Are the hazardous material and pollutants identified and kept to a minimum?
- Are the material and equipment chosen for the design costly to handle, store and to disassemble or waste?
- Are special containers or facilities necessary when using the hazardous materials or pollutants?

4.9 Human factors/ergonomics/accessibility
- Are doors provided where appropriate? Are they hinged?
- Are the size and location of openings adequate for access?
- Are the doors and openings labelled? What information is contained in the label?
− Are access doors fasteners minimized?
− Are access doors fasteners of the type quick-release?
− Can access be gained without tools?
− If tools are needed to gain access are the number minimized and of standard variety?
− Are access between modules and components adequate?
− Are the hazardous material and pollutants identified and kept to a minimum?
− Is it necessary to use protective equipment when performing the maintenance task?
− Are maintenance tasks possible to perform with protective equipment if necessary (eg gloves, helmet)?
− Are number of lifting devices for heavy or bulky items minimized?
− Is the time it takes to carry out a maintenance task reasonable compared with the working position?
− Are access requirements compatible with the frequency of maintenance?

4.10 Obsolescence
− Is the item at risk for obsolescence during the lifetime of the product?
− Is the design made to facilitate redesign if necessary due to obsolescence?
− Is there a possibility of emulation or re-creation of the design?
− Is there an after market source available?
− Is the supplier required to inform and initiate a purchase process of parts to cover the rest of the lifetime?
Chapter 6

Human factors analysis

Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human factors analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>1</td>
</tr>
<tr>
<td>1 General</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>2</td>
</tr>
<tr>
<td>2 Logistic support analysis and human factors</td>
<td>2</td>
</tr>
<tr>
<td>2.1 Human physical abilities</td>
<td>2</td>
</tr>
<tr>
<td>2.2 Limitation of human beings because of health threat</td>
<td>2</td>
</tr>
<tr>
<td>3 Human factors analysis aspects</td>
<td>3</td>
</tr>
<tr>
<td>3.1 Influence on design</td>
<td>3</td>
</tr>
<tr>
<td>3.2 LSA guidance</td>
<td>3</td>
</tr>
<tr>
<td>4 Human factors to be considered</td>
<td>4</td>
</tr>
<tr>
<td>4.1 Anthropometric aspects</td>
<td>4</td>
</tr>
<tr>
<td>4.2 Ergonomic aspects</td>
<td>4</td>
</tr>
<tr>
<td>4.3 Environmental aspects</td>
<td>4</td>
</tr>
<tr>
<td>5 Additional information</td>
<td>5</td>
</tr>
</tbody>
</table>

List of tables

1 References .................................................................................................................. 1
2 Sample limits for lifting .......................................................................................... 4
3 Sources for additional information ......................................................................... 5

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL-STD 1472F</td>
<td>Human engineering, design criteria for military systems, equipment and facilities</td>
</tr>
</tbody>
</table>

1 General

1.1 Introduction

This chapter describes the relationship and integration between human factors and the logistic support analysis process. Human factors provide source data that can be used by LSA to determine maintenance crew and support equipment requirements. This relationship begins in the design process and continues through the development of the maintenance task analysis.

1.2 Objective

The functions of LSA, maintainability and supportability respectively must be closely coordinated to ensure that potential support solutions are within established support thresholds.
including the requirements of human factors. This is accomplished by considering the crew size and the required support resources for operational and maintenance tasks. Limitations because of human factors influence the establishment of the support environment as well as the design of the Product itself. The factor human being determines the activities concerning usage of a Product and support of a Product. Especially in the area of support, human factors have decisive influence on the practicability of operational or maintenance activities.

1.3 Scope
This chapter is directed at logistics personnel who perform technical and logistic analysis activities. Some of the analysis activities are highly influenced by the abilities and the natural limitations of a human being. The results of the human factors analysis should be documented carefully and they should be available at a very early project phase. Some logistic decisions are influenced decisively by human factors. Within the LSA database the influence of human factors can appear at different places. For example warnings and cautions can be a typical criterion as well as the need for specific equipment to perform a task and to protect the human being against the unpleasing environment.

2 Logistic support analysis and human factors

2.1 Human physical abilities
The LSA and human factors integration occurs throughout the whole product life cycle. Each modification or proposed design change could require changes in maintenance which require a review of the human factor constraints and limitations. Especially at the end of the life cycle during the disposal phase, human factors can be of crucial interest because of the need of handling material, which is critical for human health.

LSA and human factors functions have a very common objective but distinct purposes or goals. The human factors aspects focus on the various standards concerning:

- Anthropometric aspects
- Ergonomic aspects
- Environmental aspects
- Other physiological aspects

LSA should take into account these standards to evaluate potential support solutions. One of many examples of this is the diameter of the average human forearm. Any access panel that requires a reach beyond the wrist must conform to the minimum size that is standardized accordingly. The human factors should influence the design to assure that this size requirement is achieved for all access panels. In addition, LSA has to consider the need for special tooling that may be required to complete a given task being performed under limited moving space conditions. During this analysis if an alternative is recognized that alternative may be presented back to design for reconsideration.

2.2 Limitation of human beings because of health threat
Another aspect concerning human factors is the limitation of human activity because of health threat. The handling of dangerous material or material which is hazardous to health must follow strict regulations to ensure physical integrity. The same holds for work which must be carried out under extreme environmental conditions like:

- very cold or very hot environment
- very humid environment
- working underground or underwater
- critical environment because of other reasons (eg dust, exposure to fumes, noise)

Regulations have to ensure the protection of human beings against the implications of these environmental impacts. These regulations must be taken into consideration for logistic analysis activities as well as the limitations because of physical restrictions.
3 Human factors analysis aspects

3.1 Influence on design
There are many industry standards that address human factors design constraints and requirements. These should be appropriately referenced as part of the developer/customer agreements and included in the contract.

The human factors qualitative data have to be taken into account for determination of the maintenance crew skills, size and support equipment needs. For example, human factors will provide weight limits for a single person lift. These weight limits are then used to identify the maintenance crew size for a specific task. Same analysis will also identify the need for mechanical lifts and/or maintenance stands. This information may be used in a trade-off study for design alternatives as an iterative process. Human factors provide specifications and standards that have to be applied to the individual maintenance tasks.

If applicable, the personnel requirement "crew size" should take into account the demographic aspects as well. For example, an all male crew may have a greater lift limit than a mixed male/female crew. Therefore, the weight of the item may define the proper number of people required for lifting but that quantity could change with the specific crew demographics. The weigh restriction is to be taken into account for mechanical lift requirements and therefore impact support equipment requirements. The location may influence the need for maintenance stands but the maintenance crew size should be considered in the definition of the type of stand required.

3.2 LSA guidance
The LSA guidance conference should identify the rules and guidelines that will be applied to the LSA and human factors analysis which may include eg maintenance crew demographics, lift, reach and other standards and limitations. The customer and contractor must agree on specific standards that will be applied. In addition to agreeing on the standard it is advisable to review the standard and document any exceptions.

LSA receive input from design and develop trade-off analysis of each drawing alternative. From the first drawing effort, LSA will analyze the design and compare support requirements between alternatives and make recommendations based on the supportability and life cycle cost. The process involves identifying all the possible maintenance actions. Each maintenance actions will result in a list of resources required to accomplish each task. These resources include part, maintenance man-hours, training, support equipment and test equipment. The specific considerations relative to human factors will be the number of maintainers and the amount of human factor related support equipment.

The number of maintainers may be influenced by the variety of skills required and the weight and size of the parts being changed during the maintenance action. Factors effective the skills will be sub-tasks that are necessary to complete the maintenance action. The weight and the height it must be lifted should also be taken into account in order to identify the maintenance crew size. Each of the various Human Factor standards has weight limits for a single person lift. The amount may vary based on crew demographics and the height it must be lifted. Also if the item is to be carried over a distance the standards should define the weight and distance limits. Table 2 below is taken from MIL-STD 1472F and is a sample of those limits reflecting US DOD regulations.

These limits may vary between standards but in all cases should be applied to the maintenance task analysis. Should the item exceed these limits then additional maintainers should be used provided the appropriate number of handles are attached. The two-hand lift defined above will require two handles or two grip areas. If multiple persons are required for lifting then the charts in the Human Factor standards that define maintainer size must be checked against the physical dimensions of the item to assure that multiple persons can actually assist in the maintenance. If the weight or size exceeds personnel lift limits then the item may require mechanical assist for lifting.
Table 2  Sample limits for lifting

<table>
<thead>
<tr>
<th>Handling function</th>
<th>Population (male and female)</th>
<th>Population (male only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lift an object from the floor and place it on a surface not greater than 152 cm (5 ft) above the floor.</td>
<td>16.8 kg (37 lb)</td>
<td>25.4 kg (56 lb)</td>
</tr>
<tr>
<td>Lift an object from the floor and place it on a surface not greater than 91 cm (3 ft) above the floor.</td>
<td>20.0 kg (44 lb)</td>
<td>39.5 kg (87 lb)</td>
</tr>
<tr>
<td>Carry an object 10 m (33 ft) or less.</td>
<td>19.0 kg (42 lb)</td>
<td>37.2 kg (82 lb)</td>
</tr>
</tbody>
</table>

The entire process is an iterative process and will need to be used for each design change. LSA will begin the more detailed maintenance task analysis once the final design is approved. The maintenance task analysis will then compare each of the maintenance subtasks against the human factors standard to assure that the proper steps are documented.

4 Human factors to be considered

The following paragraph should give a very rough overview which human factors should be taken into consideration within LSA activities. The lists are not limited to the aspects given here but should serve to receive an impression which aspects of human factors can influence the design of Products and, especially in the area of LSA, the design of the support environment (operational and maintenance related).

4.1 Anthropometric aspects
- Lines of sight (visual field, vertical and horizontal)
- Audio signals requirements
- Muscle strength of arms, hands and thumb
- Required muscle strength for vertical pull extensions
- Required muscle strength for horizontal push and pull movements
- Maximum weight of units to be lifted
- Maximum weight of support equipment
- Arm and hand access dimensions

4.2 Ergonomic aspects
- Design of controls (e.g. switches, cranks, joysticks, ball controls, hand wheels, levers, pedals, knobs)
- Minimum handle dimensions
- Workspace design (e.g. seated, standing, mobile)
- Difficult accessibility - ramps and ladders
- Doors and access panels dimensions
- Illumination requirements

4.3 Environmental aspects
- Effective temperature
- Limits of extreme cold and warm temperature conditions
- Influence of wind-chill on human beings
- Decreased performance of human beings under extreme climatic conditions
- Ventilation requirements
- Exposure limits ultraviolet radiant energy
- Exposure limits to pollution like dust, fumes, etc
- Noise limitations
5 Additional information

Each program should define a standard for use in the analysis process. The links below are examples where you can find more detailed information to evaluate the human factor related support requirements of any design.

<table>
<thead>
<tr>
<th>Source</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>US Government standards</td>
<td><a href="http://www.hfetag.com">http://www.hfetag.com</a></td>
</tr>
<tr>
<td>Human Factors and Ergonomics Society (HFES)</td>
<td><a href="http://www.hfes-europe.org">http://www.hfes-europe.org</a></td>
</tr>
<tr>
<td>Designing for humans</td>
<td><a href="http://www.designingforhumans.com">http://www.designingforhumans.com</a></td>
</tr>
</tbody>
</table>
# Chapter 7

**LSA failure modes and effects analysis**

## Table of contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSA failure modes and effects analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Interface with related analyses</td>
<td>3</td>
</tr>
<tr>
<td>1.4.1 FMECA in design and development activities</td>
<td>3</td>
</tr>
<tr>
<td>1.4.2 LSA FMEA and design-oriented FMECA</td>
<td>4</td>
</tr>
<tr>
<td>1.4.3 LSA FMEA and maintainability, maintenance and safety analysis</td>
<td>4</td>
</tr>
<tr>
<td>2 Analysis procedure and subprocess description</td>
<td>4</td>
</tr>
<tr>
<td>2.1 General flow chart</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Establish FMEA for replaceable units</td>
<td>5</td>
</tr>
<tr>
<td>2.2.1 Limit the breakdown level</td>
<td>5</td>
</tr>
<tr>
<td>2.2.2 Easy identification</td>
<td>5</td>
</tr>
<tr>
<td>2.2.3 Take software into account in the breakdown</td>
<td>6</td>
</tr>
<tr>
<td>2.2.4 Physical breakdown and LSA FMEA</td>
<td>6</td>
</tr>
<tr>
<td>2.2.5 Prediction of failure rates</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Identify available means of failure detection</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 Detection means</td>
<td>8</td>
</tr>
<tr>
<td>2.3.2 Detection rates and detectability ratings</td>
<td>8</td>
</tr>
<tr>
<td>2.3.3 Identification of requirement for new test equipment</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Identify possible means of localization of failed items</td>
<td>9</td>
</tr>
<tr>
<td>2.4.1 Localization means</td>
<td>9</td>
</tr>
<tr>
<td>2.4.2 Localization ability ratings</td>
<td>10</td>
</tr>
<tr>
<td>2.5 Analyze requirements for troubleshooting procedure</td>
<td>10</td>
</tr>
<tr>
<td>2.5.1 Cases in which a procedure is required</td>
<td>10</td>
</tr>
<tr>
<td>2.5.2 Detection confirmed upon possible false alarm</td>
<td>11</td>
</tr>
<tr>
<td>2.5.3 Interface with criticality analysis</td>
<td>11</td>
</tr>
<tr>
<td>2.5.4 Troubleshooting upon detected but not localized failure</td>
<td>12</td>
</tr>
<tr>
<td>2.6 Elements required for a troubleshooting procedure</td>
<td>12</td>
</tr>
<tr>
<td>2.7 Record results</td>
<td>12</td>
</tr>
<tr>
<td>3 Inputs</td>
<td>13</td>
</tr>
<tr>
<td>3.1 Physical breakdown</td>
<td>13</td>
</tr>
<tr>
<td>3.2 FMEA</td>
<td>13</td>
</tr>
<tr>
<td>4 Outputs</td>
<td>13</td>
</tr>
<tr>
<td>4.1 Subprocess outputs to be recorded</td>
<td>13</td>
</tr>
<tr>
<td>4.2 LSA FMEA tabular report</td>
<td>14</td>
</tr>
<tr>
<td>5 Complementary analysis</td>
<td>16</td>
</tr>
<tr>
<td>5.1 Criticality analysis</td>
<td>16</td>
</tr>
<tr>
<td>5.2 Removal criteria</td>
<td>16</td>
</tr>
<tr>
<td>5.3 Software failure analysis</td>
<td>16</td>
</tr>
<tr>
<td>5.4 Troubleshooting task analysis</td>
<td>16</td>
</tr>
<tr>
<td>5.5 New test equipment management documents</td>
<td>17</td>
</tr>
<tr>
<td>6 Lessons learned</td>
<td>17</td>
</tr>
<tr>
<td>6.1 Undetectable failures</td>
<td>17</td>
</tr>
<tr>
<td>6.2 Random failures</td>
<td>17</td>
</tr>
<tr>
<td>6.3 Influence on design</td>
<td>18</td>
</tr>
</tbody>
</table>

Applicable to: All
6.4 Failure rates and failure modes rates ................................................................. 18
6.5 Limits of software failure analysis ...................................................................... 18

List of tables

1 References ............................................................................................................. 2
2 Failure mode ratio distribution .............................................................................. 7
3 Failure mode ratings .............................................................................................. 8
4 Detection ability ratings ....................................................................................... 9
5 Localization ability ratings .................................................................................. 10
6 Troubleshooting upon false alarm ....................................................................... 11
7 Troubleshooting assessment from localization rate of built-in tests ..................... 12
8 Outputs of each sub process ............................................................................... 13
9 Recommendations for LSA FMEA tabular report ............................................. 14

List of figures

1 Logical flowchart of analysis procedure ............................................................... 5
2 From technical FMEA to LSA FMEA - grouping of failure modes ..................... 6
3 LSA FMEA tabular report .................................................................................... 14

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 3</td>
<td>LSA Business process</td>
</tr>
<tr>
<td>Chap 21</td>
<td>Terms, abbreviations and acronyms</td>
</tr>
</tbody>
</table>

1 General

1.1 Introduction

Logistic Support Analysis (LSA) Failure Modes & Effects Analysis (FMEA) is part of event-driven maintenance analysis.

Whereas Scheduled Maintenance Analysis (Chap 10) focuses on scheduled and preventive maintenance, LSA FMEA (Chap 7) and Damage and Special Event Analysis (Chap 8) focus on corrective maintenance.

More precisely, LSA FMEA establishes the methodology and decision logic that should be a prerequisite to the identification of corrective maintenance tasks to be applied to the Product in case of an inherent failure occurrence.

1.2 Objective

All items and equipment liable to fail are candidates for corrective maintenance whether they are subject to servicing or preventive maintenance tasks or not. Some failures may be detected or observed during operation or during a maintenance task or inspection. Other failures may be detected but not localized, and others are not detectable at all.
Therefore, in several cases, additional search and troubleshooting work will be necessary to precisely identify and localize failed items so that they can be removed and replaced.

This work:
- is characterized by data that could be derived from a FMEA and must be populated in the LSA record
- must be documented in the maintenance publications
- may require special tools and test equipment

The objective of this procedure is to provide a methodology that helps define the maintenance tasks rendered necessary by the inherent limits of reliability and testability (integrated or not), and the related support means.

1.3 Scope
The scope of this procedure is to:
- exploit existing documents as far as possible (FMEA from the design office, for instance), or build on existing FMEA data or build a new FMEA, tailored to LSA needs, if no existing documents are available
- analyze means available for detection of failures and for localization of failed items
- identify possible requirements for specific troubleshooting procedures
- identify possible requirements for special tools and test equipment
- record all results

It is applicable on any item within the system, specifically developed for the needs of a new system, re-used from a previous program, or acquired from the commercial-off-the-shelf range of a supplier.

1.4 Interface with related analyses
Several types of FMEA and Failure Mode, Effects and Criticality Analysis (FMECA) exist as parts of design, development and production management activities. According to the nature and requirements of the individual program, these FMECA may be:
- mandatory to satisfy regulatory requirements (e.g. nuclear plants, chemical plants, transports systems subject to airworthiness or seaworthiness regulations, military systems)
- purposely implemented to manage the achievement of an objective (e.g. improve safety, improve reliability, improve mission availability) throughout the program
- purposely implemented to identify and justify corrective maintenance activities triggered by inherent failure causes: this is the purpose of LSA FMEA

1.4.1 FMECA in design and development activities
The FMECA is an essential function in design from concept through development. The effective FMECA is iterative and corresponds with the nature of the design process itself. The extent of effort and sophistication of approach used in the FMECA depends upon the nature and requirements of the individual program. This makes it necessary to tailor the requirements for a FMECA to each individual program. Tailoring requires that, regardless to the degree of sophistication, the FMECA contributes meaningfully to program decision. A properly performed FMECA is invaluable to those who are responsible for making program decisions regarding the feasibility and adequacy of a design approach.

The usefulness of the FMECA as a design tool and in the decision making process depends upon the effectiveness with which problem information is communicated for early design attention. Timeliness is one of the most important factors in differentiating between effective and ineffective implementation of FMECA. While the objective of a FMECA is to identify all modes of failure within a system design, its first purpose is the early identification of all catastrophic and critical failure possibilities so they can be eliminated or minimized through design correction at the earliest possible time. Therefore, the FMECA is generally initiated as a functional FMECA.
as soon as preliminary design information is available at the higher system levels, and extended to the lower levels as more information becomes available on the items in question.

Although the FMECA is an essential reliability task for design activities, it also provides information for other purposes. The use of the FMECA or of some analysis derived from it is called for in maintainability, safety analysis, availability analysis, logistic support analysis, elaboration of maintenance concept, and in failure detection and fault isolation. This coincident use must be a consideration in planning the FMECA effort to prevent the harmful proliferation of requirements and the duplication of efforts within the same contractual program.

One of the analyses derived from FMECA is LSA FMEA.

1.4.2 LSA FMEA and design-oriented FMECA
Whereas design-oriented FMECA needs to analyze and quantify the reliability of each individual component of a system, such a level of detail not always essential for maintenance purposes because the maintenance activities may not focus on the individual components but on replaceable or repairable units, which maybe located at a higher level of breakdown than that of individual components.

Therefore the LSA FMEA is generally coincident with, but not identical to, design-oriented FMECA. Tight interface, coordination and milestones must be implemented between these two activities to ensure adequate consistency, timeliness and traceability between them. The interfaces and specific data element needs shall be defined in the LSA Plan, and agreed between LSA, design FMEA functions and testability analysis.

1.4.3 LSA FMEA and maintainability, maintenance and safety analysis
Other maintainability, maintenance or safety analyses use data from LSA FMEA or derived from FMECA, eg Level of Repair Analysis (LORA), Reliability-Centered Maintenance (RCM) and Scheduled Maintenance Analysis (SMA). In order to take full benefit of existing works and to avoid useless duplication of works, these other activities must be tightly interfaced and coordinated with FMECA and LSA FMEA in the LSA Program Plan, refer to Chap 3.

2 Analysis procedure and subprocess description
2.1 General flow chart
The workflow of the analysis procedure is presented on the general flowchart hereafter.
2.2 **Establish FMEA for replaceable units**

As a general consideration, the process described hereafter is based on the re-use, and possible additional data elements to be added, of an existing FMEA/FMECA. These results should be grouped as much as feasible in order to keep the effort to be spent at a sensible, low level. This sub-process essentially consists of merging the FMEA/FMECA and the physical breakdown. Preparatory works may be necessary before this merging is rendered possible.

2.2.1 **Limit the breakdown level**

The physical breakdown generally goes down to individual components required for detailed definition purposes. For the needs of malfunctioning analysis, it does not need to be developed beyond the level of replaceable units, except if further development of breakdown helps in understanding or describing failure modes and/or possible inspection or repair of the replaceable unit. A first step is to limit the physical breakdown at the appropriate level. This is usually done by extraction and simplification of the complete physical breakdown.

2.2.2 **Easy identification**

Failures, failure modes and items of the physical breakdown will be frequently quoted in further analysis and analysis reports. It is recommended to adopt an easy-to-use identification method for these failures, failure modes and items.
2.2.3 Take software into account in the breakdown
Replaceable units that have software should include this software as sub-component for configuration management purposes. This is independent of the reload ability of software, provided that reload ability is equivalent for software of replace ability for hardware. Note that some software may have defects, too, especially on non-mature systems.

2.2.4 Physical breakdown and LSA FMEA
LSA FMEA must be compliant with the physical breakdown. This means the items liable to fail must be taken among the items of the physical breakdown. This also means that failures must be listed or grouped for each replaceable unit.

Fig 2 From technical FMEA to LSA FMEA - grouping of failure modes
Grouping should be performed by taking into consideration the following criteria:

− All failure modes generated within a technical FMEA/FMECA, which cause one and the same chain of maintenance activities, should be grouped to one LSA failure mode within the LSA FMEA.
− All aspects which have to be taken into consideration concerning grouping of failure mode should be agreed between customer and contractor within the LSA guidance conference. Fig 2 gives an overview how these crucial grouping criteria can be realized.

While grouping, the failure rate of the item under analysis will be distributed to the LSA failure modes. This is required for calculation of the corresponding task frequencies of the rectifying tasks linked to the LSA failure modes.
Table 2 Failure mode ratio distribution

<table>
<thead>
<tr>
<th>Failure Mode</th>
<th>FMR</th>
<th>Rectifying tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSA failure mode 1</td>
<td>0,10 (10%)</td>
<td>Repair procedure 1 + Failure detection procedure 1</td>
</tr>
<tr>
<td>LSA failure mode 2</td>
<td>0,15 (15%)</td>
<td>Repair procedure 1 + Failure detection procedure 2</td>
</tr>
<tr>
<td>LSA failure mode 3</td>
<td>0,50 (50%)</td>
<td>Repair procedure 2 + Failure detection procedure 1</td>
</tr>
<tr>
<td>LSA failure mode 4</td>
<td>0,25 (25%)</td>
<td>Repair procedure 3 + Failure detection procedure 3</td>
</tr>
</tbody>
</table>

The Failure Mode Ratio (FMR) identifies the fraction of the individual failure mode on the entire failure rate of the item under analysis.

**Note**

It must be considered that the distribution of failure rates in connection with physical breakdown elements must be analyzed from breakdown indenture level to the next neighbouring indenture level. This principle is of crucial importance to get a coherent view of failure rates within a complete physical breakdown.

In practice, the breakdown is often incomplete down to the last small part. In this case, it can be possible that the sum of failure rates of indenture level \( n+1 \) is smaller than the failure rate of the breakdown element on indenture level \( n \). On the other side, a failure rate on indenture level \( n \), which is smaller than the sum of failure rates on the deeper indenture level \( n+1 \) is not logical and should be rechecked carefully.

In the case of an idealized situation of a complete breakdown, the sum of the failure rates of indenture level \( n+1 \) should be the failure rate of the breakdown element of indenture level \( n \).

**2.2.5 Prediction of failure rates**

For the purposes of LSA FMEA, failure rates must be established for each LSA failure mode.

If no analytical method is applicable, the failure rates may be assessed by best engineering judgement from experience on similar systems of comparable complexity and operated in a similar context.

Failure rates may be calculated by an analytical method (eg MIL HDBK 217, RDF-93, RDF-2000 UTE-C 80-180).

If no dependable failure rate is available, a rough rating table should be established. The following qualitative table is provided as an example. It can be tailored according to needs and to the operating context, which must be defined first.
### Table 3 Failure mode ratings

<table>
<thead>
<tr>
<th>Rating</th>
<th>Occurrence</th>
<th>Description</th>
<th>Probability of single failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Very low likelihood</td>
<td>Very few failures likely</td>
<td>Less than 0.001</td>
</tr>
<tr>
<td>2</td>
<td>Low likelihood</td>
<td>Few failures likely</td>
<td>Between 0.001 and 0.01</td>
</tr>
<tr>
<td>3</td>
<td>Moderately low likelihood</td>
<td>Occasional failures likely</td>
<td>Between 0.01 and 0.10</td>
</tr>
<tr>
<td>4</td>
<td>Medium likelihood</td>
<td>Medium number of failures likely</td>
<td>Between 0.10 and 0.20</td>
</tr>
<tr>
<td>5</td>
<td>Moderately high likelihood</td>
<td>Moderately high number of failures likely</td>
<td>Higher than 0.20</td>
</tr>
</tbody>
</table>

**Note**

The above ratings 1 to 5 correspond respectively to the levels E to A of the qualitative approach of MIL-STD-1629 Procedure for performing a failure mode, effects and criticality analysis.

### 2.3 Identify available means of failure detection

#### 2.3.1 Detection means

For each possible failure of a replaceable unit:

- Check whether the failure is detected by any form of automatic detection which triggers a warning device.
- Check whether the failure is liable to be detected by a BIT. If yes, assess the detection rate and false alarm rate of this BIT. These rates are relevant here when related to replaceable units, not to functions.
- Irrespective of the answer to the previous question, check functional symptoms and/or physical symptoms likely to help detect the failure. For this check, failure effects listed in FMEA may be considered as a guideline.
- List all failures detectable by any of the above means, record the associated detection means.
- List failures undetectable by any of the above means.

**Note**

These detection means are a generalization, irrespective of the level of maintenance, of the failure detection methods and failure detection means of MIL-STD-1629.

#### 2.3.2 Detection rates and detectability ratings

The detection rate is the probability of the detection means to effectively detect the failure.

If some built-in tests (BIT) are embedded in the replaceable units, detection rates may be assessed by an analytical method as part of testability analysis. Therefore the LSA FMEA should be adequately interfaced with testability analysis in order to take benefit of the detection rates issued from it. This data may not be available from the Technical FMEA/FMECA.

If no analytical method is applicable, a qualitative approach should be preferred. This qualitative assessment may be based on best engineering judgment. The following table is provided as an example. It must be tailored according to needs.
Table 4 Detection ability ratings

<table>
<thead>
<tr>
<th>Rating</th>
<th>Detection probability</th>
<th>Description</th>
<th>Detection rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High likelihood</td>
<td>High effectiveness for detection</td>
<td>Higher than 80%</td>
</tr>
<tr>
<td>2</td>
<td>Moderately high likelihood</td>
<td>Moderately high effectiveness for detection</td>
<td>Between 80% and 60%</td>
</tr>
<tr>
<td>3</td>
<td>Medium likelihood</td>
<td>Has medium effectiveness for detection</td>
<td>Between 60% and 40%</td>
</tr>
<tr>
<td>4</td>
<td>Moderately low likelihood</td>
<td>Moderately low effectiveness for detection</td>
<td>Between 40% and 20%</td>
</tr>
<tr>
<td>5</td>
<td>Low likelihood</td>
<td>Low effectiveness for detection</td>
<td>Less than 20%</td>
</tr>
</tbody>
</table>

Note
The probability of the detection means to effectively detect the failure must be rated in accordance with what is considered effective in terms of scheduled maintenance analysis. That means that the above ratings must be consistent with the effectiveness of visual inspections, detailed inspections, special detailed inspection, visual checks, operational checks and functional checks deemed effective for the purpose of scheduled maintenance.

2.3.3 Identification of requirement for new test equipment
The above table may be used to identify a requirement for additional test equipment or built-in-test liable to raise detection probability from an unacceptable rating up to an acceptable one.

This identification is generally performed as part of testability analysis for BIT’s, and as part of maintenance task analysis for additional test equipment. Therefore the use of the above table for the purpose of identifying a requirement for an additional detection means should be interfaced with testability analysis and with Maintenance Task Analysis (MTA).

2.4 Identify possible means of localization of failed items
2.4.1 Localization means
For each detected failure of a replaceable unit:

- Check whether an automatic detection (if any) triggering a warning device is likely to localize without ambiguity the one replaceable unit that has failed, or if it is likely to localize one out of n replaceable units that has failed (one-among-n ambiguousness).
- Check whether a BIT (if any) is likely to localize without ambiguousness the one replaceable unit that has failed, or if it is likely to localize one out of n replaceable units that has failed (one-among-n ambiguousness).
- Check whether a monitoring system is likely to localize without ambiguousness the one replaceable unit that has failed, or if it is likely to localize one out of n replaceable units that has failed (one-among-n ambiguousness). There are such monitoring systems for health and usage monitoring, for example.
- Check whether applicable functional checks, leading to detection by functional symptom, are likely to localize without ambiguousness the one replaceable unit that has failed, or if they are likely to localize one out of n replaceable units that has failed (one-among-n ambiguousness).
- Check whether cross-checking of BIT results may help reduce the ambiguousness of the localization (i.e. reduce the value of n in the "one-among-n ambiguousness"). For this cross-check, a special attention may be paid to BIT results which are impacted by a common cause.
Check whether applicable visual checks or measurements without any dismounting or removal, leading to a detection by physical symptom, is likely to localize without ambiguity the one replaceable unit that has failed, or if it is likely to localize one out of $n$ replaceable units that has failed (one-among-$n$ ambiguousness).

### 2.4.2 Localization ability ratings

Localization is limited to detected failures.

Localization ability ratings may be elaborated as part of testability analysis on products entirely covered by built-in tests. Some additional engineering may be required to the results of these tests have to be cross-checked with other source of information (eg functional checks, visual checks, visual inspections without removal) in order to reduce the ambiguity. In some cases, this additional engineering may be performed by an activity external to LSA (eg fault tree analysis); if it is the case, adequate interface must be implemented with LSA FMEA. This data may not be available from the technical FMEA/FMECA. If that is the case the data should be retrieved from the testability analysis.

The following table is provided as an example. It must be tailored according to needs.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Localization probability</th>
<th>Description</th>
<th>Localization ability rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High likelihood</td>
<td>High effectiveness of the abovementioned localization means</td>
<td>Higher than 90%</td>
</tr>
<tr>
<td>2</td>
<td>Moderately high likelihood</td>
<td>Moderately high effectiveness for localization</td>
<td>Between 80 and 90%</td>
</tr>
<tr>
<td>3</td>
<td>Medium likelihood</td>
<td>Medium effectiveness for localization</td>
<td>Between 60% and 80%</td>
</tr>
<tr>
<td>4</td>
<td>Moderately low likelihood</td>
<td>Moderately low effectiveness for localization</td>
<td>Between 40% and 60%</td>
</tr>
<tr>
<td>5</td>
<td>Low likelihood</td>
<td>Low effectiveness for localization</td>
<td>Below 40%</td>
</tr>
</tbody>
</table>

### 2.5 Analyze requirements for troubleshooting procedure

#### 2.5.1 Cases in which a procedure is required

Basically, a troubleshooting procedure is required when:

- a failed item is not likely to be localized without ambiguity by any of the localization means described in Para 2.4.
- a failure has been detected by a built-in test known to have a high false alarm rate.

However, cases in which such a procedure is required are hardly ever defined by a simple cross-grid of the previous tables. The main reasons for this are:

- detection rate and localization rate are statistical variables, and are generally bound to failure rate and failure mode rate.
- defining a threshold on the detection and the localization rates would create an artificial threshold effect.
- if the failure has been detected by a built-in test with a false alarm rate known to be high, its results may not be ignored if the failure has some possible impact on safety. This remains true even if the BIT has been repeated without detecting a failure.
the requirement for troubleshooting is not entirely dictated by detection and localization ability and should be analyzed in the light of complementary elements, such as accessibility, removal and replacement costs, or safety.

Therefore, analyzing the requirement for a troubleshooting procedure needs some expert's best engineering judgment. The following guidelines may be used after being tailored according to needs. They assume that detection rate and localization rate take the failure rate/failure mode rate into account.

2.5.2 Detection confirmed upon possible false alarm

Prior to any troubleshooting, the failure detection may be confirmed if there is a doubt about the reality of the detected failure. However, it must be remembered that a non-repeated detection of failure may be due to a random failure of the tested equipment itself, regardless of false alarm rate of the testing device.

Therefore, it is recommended to assess the need for this preliminary step from the importance of the consequences of the failure:

<table>
<thead>
<tr>
<th>Consequences of failure</th>
<th>Repetition of test for removal of ambiguousness</th>
<th>Follow-on actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negligible</td>
<td>The test may be repeated a certain number of times (to be determined) with the usual test device or built-in test until it is confirmed or invalidated.</td>
<td>If the failure detection is invalidated, no further analysis is to be carried out and the equipment may be declared back to operational status. If the failure is confirmed, need for troubleshooting is assessed as in Para 2.5.4.</td>
</tr>
<tr>
<td>Not negligible but acceptable</td>
<td>The test may be repeated a certain number of times (to be determined) with an independent test device until it is confirmed or invalidated.</td>
<td>If the failure detection is invalidated, no further analysis is to be carried out and the equipment may be declared back to operational status. If the failure is confirmed, need for troubleshooting is assessed as in next paragraph.</td>
</tr>
<tr>
<td>Unacceptable</td>
<td>The test needs not be repeated.</td>
<td>Proceed as in next paragraph.</td>
</tr>
</tbody>
</table>

In any case, the need for repetition of test and the follow-on actions must be documented in a procedure which should be applicable prior to the troubleshooting procedure itself.

2.5.3 Interface with criticality analysis

On products on which a criticality analysis is performed, thresholds of acceptability used in the left column of the above table must be consistent with the consequences of the failure mode and/or the criticality of the replaceable item.

In this approach, the worst potential consequence of the failure must be taken in consideration and determined by the degree of injury, property damage, system damage or environmental impacts that could ultimately occur. This analysis is addressed by the severity analysis in FMECA methods (eg MIL-STD-1629).
This potential consequence of a failure mode may or may not be combined with the probability of occurrence of the failure, depending on the nature of the product. This combination is addressed by the criticality analysis in FMECA methods (eg MIL-STD-1629).

The failure on critical or very severe failure modes should always be considered as unacceptable, even if there is a possibility for the detection to be a false alarm. Therefore, the troubleshooting procedure requirement analysis must be interfaced with the FMECA.

2.5.4 Troubleshooting upon detected but not localized failure
The following table is just provided as a guideline. It should be tailored according to needs.

<table>
<thead>
<tr>
<th>Localization rating</th>
<th>Troubleshooting required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No troubleshooting required. Localization will be performed by the BIT.</td>
</tr>
<tr>
<td>2</td>
<td>Troubleshooting is required but may be performed by cross-checking of BIT results.</td>
</tr>
<tr>
<td>3</td>
<td>Troubleshooting is required and may require an intervention of maintenance team to perform additional research of symptoms</td>
</tr>
<tr>
<td>4 or 5</td>
<td>Troubleshooting is required, using test equipment likely to compensate the inability of the BIT to localize the failed replaceable unit.</td>
</tr>
</tbody>
</table>

2.6 Elements required for a troubleshooting procedure
Elaborating on the troubleshooting procedure itself is out of the scope of LSA FMEA. However, in order to simplify subsequent works, several documents or analyses used to determine if such a procedure is required should be noted for further use:

- The technical documents and definition files that must be used to analyze the troubleshooting requirement (eg technical plan, wiring diagram, interface description).
- Logical order of the tests to be carried out until the failed replaceable unit is identified and localized. This logical order may be backed-up by a flow-chart diagram on which all decisions should be applicable (that means, associated with the clear interpretation of a symptom or a BIT result). All decision not associated with such an interpretation should be considered as not applicable. Test equipment or inspection procedures which may be necessary should be documented.
- Possible cross-checks that could help identify and localize the failed unit. Such cross-checks are fruitful for identifying sources of common cause failures. They should concern items exposed to the effects of the common cause failure.
- Symptoms to be monitored or recorded during these cross-checks.
- Measurements to be carried out to check the physical and/or functional characteristics of the replaceable units.
- Test equipment required by the above measurements.
- Expected values for each of these measurements, and signification of unexpected measures.

Refer to Para 5.4 for the elaboration of the troubleshooting procedure itself.

2.7 Record results
Results are to be recorded as troubleshooting tasks, which are preliminary tasks for corrective maintenance. These tasks are to be attached to the next higher assembly including all the replaceable units concerned. They shall list all the support elements which have been identified during the above sub process in addition to usual data characterizing a maintenance task. The
corrective maintenance task for removal and replacement of the failed replaceable unit is to be mentioned. Troubleshooting task and removal/replacement tasks may be subtasks of a repair task.

3 Inputs

3.1 Physical breakdown
Physical breakdown (or product breakdown) provides an indented decomposition of the main product. For FMEA purposes, it should be developed down to the level of replaceable units at least. The corresponding depth of indenture may vary in accordance with the concerned item. Further development to sublevels of these items may be necessary only when it helps characterize the failure modes of the item or the related failure effects. Otherwise it can be dispensed with. Replaceable units should be easily identifiable in this breakdown.

3.2 FMEA
Failure Modes and Effects Analysis (FMEA) provides some or all of the following information:
- a list of items likely to fail
- the associated failure rates
- the related failure modes foreseen
- the probability of occurrence of these failure modes
- the consequences (effects) of these failure modes regarding the availability of the replaceable unit and its impacts on safety and overall availability
- the signature of the failure (e.g., physical symptoms, functional symptoms, BIT results)

This FMEA may be carried out as part of a product safety analysis, of a product mission reliability analysis, or provided by the supplier of the equipment.

4 Outputs

4.1 Subprocess outputs to be recorded
The outputs listed in the following table are summarized in Fig 3.

<table>
<thead>
<tr>
<th>Sub-process</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish FMEA for replaceable unit</td>
<td>Replaceable units, Possible failures, Failures modes, Failure effects</td>
</tr>
<tr>
<td>Identify available means of failure detection</td>
<td>Warnings, Built-in tests reports, Physical symptoms, Functional symptoms, Requirement for new test equipment</td>
</tr>
<tr>
<td>Identify possible means of localization of failed item</td>
<td>Warnings, Built-in tests reports, Ground test equipment, Requirement for new test equipment</td>
</tr>
<tr>
<td>Analyze requirements for troubleshooting procedure</td>
<td>Answer yes/no to need for a troubleshooting, If yes, inputs for troubleshooting task analysis</td>
</tr>
</tbody>
</table>
4.2 LSA FMEA tabular report

It is recommended to issue a tabular report that provides all important information and outputs in a convenient form.

The columns of this table may be populated as sub-processes go along.

An example of LSA FMEA tabular report is given in Fig 3. Recommendations for filling the columns are provided in Table 9.

Note

Header possibly required for purposes of traceability and quality management of the LSA FMEA has not been included. This header may provide release date, issue number, author’s name, approval signatures, information required for identification of the item/replaceable unit, etc.

This tabular report is to be tailored according to needs.

Numbers in circles refer to recommendations provided in the table on the following page.

<table>
<thead>
<tr>
<th>Replaceable unit identification</th>
<th>Function</th>
<th>Failure Modes</th>
<th>Failure causes</th>
<th>Failure effects</th>
<th>Failure mode ratio (FMR)</th>
<th>Detectability rating</th>
<th>Detection means</th>
<th>Localize ability rating</th>
<th>Localization means</th>
<th>Troubleshooting required</th>
<th>Procedure requirement</th>
<th>Test equipment required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
</tbody>
</table>

ICN-B6865-S3000L0083-001-01

Fig 3 LSA FMEA tabular report

Table 9 Recommendations for LSA FMEA tabular report

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | Replaceable unit identification  
Item name. This item may be followed by its physical breakdown, provided that each line contains only one item or sub-part of item. The item name may be accompanied with or replaced by appropriate reference number or breakdown element identifier. |
| 2      | Function  
Function that the replaceable unit fulfils or contributes to |
| 3      | Failure modes  
Cause-to-effect phenomena leading to the failure |
| 4      | Failure causes  
Triggering event or circumstances of the failure mode |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| 5      | **Failure effects**  
|        | Impacts and consequences of the failure, in terms of safety, function availability, collateral damage, non-compliance with regulations or specifications. This column is to be used to document effects of a LSA failure mode.  
|        | Failure effects may be sorted and analyzed as:  
|        | - **Local effects**: impacts of the failure on the operation and functions of the replaceable unit in the breakdown level under consideration. The impacts include second-order effects resulting from failure. The purpose of this characterization of local effects in LSA FMEA is to help define local detection criteria of failures.  
|        | - **Next higher effects**: impacts of the failure on the operation and functions of the replaceable unit in the next higher breakdown level. The purpose of this characterization of local effects in LSA FMEA is to help define detection criteria of failures.  
|        | - **End effects**: total effect that the failure has on the operation, function, or status of the Product. The end effect may be the result of a double failure. For example, failure of a redundant device may result in an end effect in the event that both the prime device and its backup (redundant) device fail; otherwise the end effect is a reduction of reliability or safety levels due to the loss of redundancy. The end effects resulting from a double failure shall be indicated in the LSA FMEA tabular report.  
|        | If necessary, Column 5 may be split in sub-columns to clarify the presentation of the above failure effects.  
| 6      | **Failure rate**  
|        | Frequency with which the replaceable unit fails, expressed in failures per hour for each failure mode  
| 7      | **Failure mode ratio or failure mode rating**  
|        | LSA FMR derived from Technical FMEA or failure mode rating rated as in Para 2.4.2.  
| 8      | **Detect ability rating**  
|        | Detection rate as assessed in Testability analysis (if any and if appropriate) or rated as in Para 2.3.2.  
| 9      | **Detection means**  
|        | Short description of detection means available for this failure. These means may be used during normal operation of the system (warning on control panel, abnormal behavior of the system, perceptible loss of function, functional symptom) or during a maintenance activity (visual inspection, post-incident checks, maintenance task, exploitation of BIT reports, physical symptoms).  
| 10     | **Localize ability rating**  
|        | Localization rate as assessed in Testability analysis (if any and if appropriate) or rated as in Para 2.4.2.  
| 11     | **Localization means**  
|        | Short description of means available for localizing the replaceable unit or group of replaceable units likely to have failed: warnings, built-in test reports (especially Initiated BIT which may not be initiated during normal operation of the system, use of ground test equipment). This column may be used to document the signature of the failure.
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td><strong>Troubleshooting required</strong> Yes or no, in accordance with Para 2.5.</td>
</tr>
<tr>
<td>13</td>
<td><strong>Procedure requirement</strong> Short summary or title of the maintenance task. This procedure may be either an existing maintenance task (if no troubleshooting is required because failed unit has been localized without ambiguity, for instance) or a troubleshooting procedure as described in Para 5.4.</td>
</tr>
<tr>
<td>14</td>
<td><strong>Test equipment required</strong> Short functional description of any additional test equipment found necessary or helpful to simplify, shorten or facilitate the troubleshooting. This test equipment may be Built-in Test Equipment (BITE) or separate equipment.</td>
</tr>
</tbody>
</table>

### 5 Complementary analysis

#### 5.1 Criticality analysis

If applicable and required, a criticality analysis should be carried out to identify critical items, for which failure would result in unacceptable impact. This analysis would support identification of scheduled maintenance action that would preclude the critical failures from occurring. Refer to Chap 10 Scheduled Maintenance Analysis (SMA).

After this scheduled maintenance analysis is performed, no critical item should be left with undetectable possible failure.

Adequate identification means must be implemented to provide adequate traceability between critical items and drawings, manufacturing, provisioning, technical publication and training. This is normally part of System Safety Analysis, which should be interfaced with LSA FMEA in order to review their consistency.

#### 5.2 Removal criteria

For items exposed to progressive deterioration or to wear-out, removal criterion must be defined in terms of:

- parameter to inspect or to monitor
- threshold for removal

#### 5.3 Software failure analysis

This analysis is complementary to physical FMEA and should identify software components liable to be bugged or defective and should also identify the possible losses of functions and/or apparent failures of the item (hardware and software) that may be induced by the defects of the software. Refer to Chap 13, Software Support Analysis (SSA).

#### 5.4 Troubleshooting task analysis

This work is carried out as any other maintenance task analysis and identifies detailed list of tools, spares, test equipment, preliminary tasks to perform for accessibility or safety, ingredients and consumables, repair validation requirements.

This procedure will identify:

- The technical documents that has to be used to perform the troubleshooting (eg technical plan, wiring diagram, interface description). These documents will be included, quoted or summarized in the troubleshooting work card.
- Logical order of the tests to be carried out until the failed replaceable unit is localized.
Possible cross-checks that could help identify and localize the failed unit.
- Symptoms to be monitored or recorded during these cross-checks.
- Measurements to be carried out to check the physical and/or functional characteristics of the replaceable units.
- Test equipment required by the above measurements.
- Expected values for each of these measurements, and signification of unexpected measures.
- Skills and manpower required for the above cross-checks and measurements.
- Possible requirement for a specific facility (e.g., dark room, dust-free atmosphere) in which the troubleshooting must be carried out.
- Recommended maintenance level (if applicable) at which the troubleshooting must be carried out.
- Removal criteria or repair criteria of each replaceable unit, if they are not already quoted in the corresponding maintenance procedures.
- Reference of maintenance procedures that are to be applied to fix, repair or replace the failed item.

5.5 New test equipment management documents
Any test equipment or built-in-test deemed necessary to improve detection and/or localization should be:
- described in a specification,
- managed through a development and qualification plan,
- adequately milestone so that the test equipment can be taken into account in the troubleshooting task analysis and in the maintenance procedure

6 Lessons learned
6.1 Undetectable failures
Special attention must be given to troubleshooting analysis of items:
- exposed to failures undetected (or hardly detectable) during operation or routine maintenance
- whose failure may have hazardous effects

For these items, appropriate periodical inspection or check should be implemented in the preventive maintenance and/or inspection plan.

For failures with catastrophic effects (i.e., more severe than hazardous) and, in some cases, for failures with hazardous effects, this may be performed as part of the scheduled maintenance analysis (SMA, refer to Chap 10). If it is the case, no special attention is to be paid to undetectable failures during LSA FMEA process. However, LSA FMEA and SMA must be adequately interfaced to ensure consistency of definitions, completeness of the failure modes taken into account, and timeliness of analyses.

6.2 Random failures
Random failures may, in some cases, not be detected again during the troubleshooting.

This may be incorrectly interpreted as a false alarm generating erroneous failure detection on a failure-free item. This interpretation is flawed, given that a failure actually occurred, even if it is not detected again on replay.

The particular form of troubleshooting induced by random failures may generally not be addressed by FMEA and is therefore out of the scope of this LSA FMEA procedure. It may be addressed and managed through a specific procedure established on a case-by-case basis upon detection of such a failure.
6.3 Influence on design
If discovered soon enough in the design process, unsolvable troubleshooting may lead to propose modifications of design, such as:

− adding test sockets
− enhancing accessibility so that a given test equipment or tooling may be used for troubleshooting
− improving testability and/or detectability

Such possible influence on design should be appropriately managed and milestone in the design process, especially if a requirement for new built-in-test or test socket is deemed necessary to improve detection and/or localization (refer to Para 2.3.3).

6.4 Failure rates and failure modes rates
The failure rates and failure modes rates possibly provided by the FMEA may be used to assess which failure cause or failure mode is the most liable to occur leading to the failure. This may be useful to propose an order of priority among the checks and tests to be carried out as part of the troubleshooting process.

However, it should be reminded that in several cases this order of priority is dictated by accessibility reasons or by costs.

6.5 Limits of software failure analysis
This analysis, if present, usually does not help rating the failure occurrence probability.

In some cases, this rating may be assessed from Quality Insurance levels applied during the development of the software, or from knowledge and confidence level placed in the provider.

Moreover, a software failure is generally not associated with a failure rate, given that software failures are not time-related. Software failures, if any, generally occurs during initial deployments of the system and are progressively corrected through a maturity plan or a product quality improvement plan. Software failure is therefore often considered as a defect: after it is corrected, it never happens again.

For these reasons, software failure may be considered as highly unlikely (or better) on mature systems. Support and maturation of software are addressed in Chap 13, Software Support Analysis.
Chapter 8

Damage and special event analysis

Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Damage and special event analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1. General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Terms, abbreviations and acronyms</td>
<td>2</td>
</tr>
<tr>
<td>2. General flowchart</td>
<td>3</td>
</tr>
<tr>
<td>3. Sub process description/analysis procedure</td>
<td>3</td>
</tr>
<tr>
<td>3.1 Identification of special events</td>
<td>3</td>
</tr>
<tr>
<td>3.2 Identify potential damage to be considered</td>
<td>5</td>
</tr>
<tr>
<td>3.3 Define element or part of the product concerned</td>
<td>6</td>
</tr>
<tr>
<td>3.4 Criticality analysis</td>
<td>7</td>
</tr>
<tr>
<td>3.5 Maintenance task requirement identification</td>
<td>7</td>
</tr>
<tr>
<td>4. Inputs</td>
<td>7</td>
</tr>
<tr>
<td>4.1 Physical breakdown</td>
<td>7</td>
</tr>
<tr>
<td>4.2 Product usage description sheet</td>
<td>7</td>
</tr>
<tr>
<td>4.3 Statistical elements on special events</td>
<td>7</td>
</tr>
<tr>
<td>4.4 Product definition</td>
<td>7</td>
</tr>
<tr>
<td>5. Outputs</td>
<td>7</td>
</tr>
<tr>
<td>5.1 Special event and damage analysis tabular report</td>
<td>7</td>
</tr>
<tr>
<td>5.2 Complementary analysis</td>
<td>9</td>
</tr>
<tr>
<td>5.2.1 Influence on design</td>
<td>9</td>
</tr>
<tr>
<td>5.3 Lessons learned</td>
<td>9</td>
</tr>
</tbody>
</table>

List of tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 References</td>
<td>2</td>
</tr>
<tr>
<td>2 Terms, abbreviations and acronyms</td>
<td>2</td>
</tr>
<tr>
<td>3 Causes of events</td>
<td>3</td>
</tr>
<tr>
<td>4 Probability of occurrence of special events</td>
<td>4</td>
</tr>
<tr>
<td>5 Technology behavior knowledge rating</td>
<td>5</td>
</tr>
<tr>
<td>6 Technology sensitivity rating</td>
<td>6</td>
</tr>
<tr>
<td>7 Recommendations for damage and event analysis tabular report</td>
<td>8</td>
</tr>
</tbody>
</table>

List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 General flow chart of damage and special event analysis application logic</td>
<td>3</td>
</tr>
<tr>
<td>2 Technology/damage evaluation rating</td>
<td>6</td>
</tr>
<tr>
<td>3 Damage and event analysis tabular report</td>
<td>8</td>
</tr>
</tbody>
</table>
References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 12</td>
<td>Maintenance task analysis</td>
</tr>
<tr>
<td>Chap 21</td>
<td>Terms, abbreviations and acronyms</td>
</tr>
</tbody>
</table>

1 General

1.1 Introduction

Special events and damages caused during normal operation of the product will result in the requirement for maintenance as defined by the Maintenance Task Analysis (MTA) (refer to Chap 12). This chapter deals with the identification and quantification of special events and damages.

1.2 Objective

It is the objective of this chapter to present a methodology for identifying and justifying the maintenance tasks appropriate when a special event or a damage occurs to the product.

Inservice feedback on similar products in operation can be used to directly identify maintenance requirements for special events and damages that occur on the new product. Use of this information can shorten the analysis process contained in this chapter.

1.3 Scope

This chapter deals with special events and damages that may occur during the service life of a product and the susceptibility of new technology to damage.

1.4 Terms, abbreviations and acronyms

The following specific terms, abbreviations and acronyms used within this chapter. More special terms are explained directly within the appropriate paragraphs. Refer to Chap 21 for a full listing of Terms, abbreviations and acronyms.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Damage</td>
<td>A loss or reduction of functionality, excluding inherent failure (intrinsic reliabilities). Normally a maintenance task will be required. Damages can be grouped into damage families. For example concerning structures, typical damages can be identified like scratches, dents or cracks. These damage families are typical candidates for a standard repair procedure.</td>
</tr>
<tr>
<td>External cause</td>
<td>A cause is said to be external when an event independent of product usage occurs.</td>
</tr>
<tr>
<td>Internal cause</td>
<td>A cause is said internal when it comes from product usage by itself.</td>
</tr>
<tr>
<td>Special event</td>
<td>An event that occurs during a Product's life that cannot be considered as a normal way of operation. It can be due either to external causes (eg meteorological phenomenon) or to out of bound use (eg over-G maneuver of an aircraft).</td>
</tr>
</tbody>
</table>
2 General flowchart
The analysis logic is presented on the general flowchart in Fig 1.

![General flowchart diagram](ICN-B6865-S3000L0068-001-01)

Fig 1 General flow chart of damage and special event analysis application logic

3 Sub process description/analysis procedure
3.1 Identification of special events
This subprocess is based on a simple product usage analysis. It consists of identifying the different possible causes and analyzing them through each phase of the product's life to determine whether they can lead to a special event or not. Identify the relevant special events, using the following methodology:

**Step 1:** List the different causes that can produce an event.

Events can be due to external or internal causes and affected by natural phenomenon or humankind. Table 3 provides examples of different types of causes. A more complete table should be developed for each project.

<table>
<thead>
<tr>
<th>Types of causes</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>External causes</td>
<td>Natural phenomenon</td>
</tr>
<tr>
<td></td>
<td>Meteorological</td>
</tr>
<tr>
<td></td>
<td>Animal</td>
</tr>
<tr>
<td></td>
<td>Stones, trees, etc.</td>
</tr>
<tr>
<td>Caused by humankind</td>
<td>Combat threat</td>
</tr>
<tr>
<td></td>
<td>Material maneuver</td>
</tr>
<tr>
<td>Caused by operation environment</td>
<td>Electromagnetic field</td>
</tr>
<tr>
<td></td>
<td>Salt/ sand/pollution laden atmosphere</td>
</tr>
<tr>
<td>Caused by transport and storage</td>
<td>Shocks, movements and vibrations</td>
</tr>
<tr>
<td>conditions (sea, air, truck, rail,</td>
<td>Depressurization</td>
</tr>
<tr>
<td>etc)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3 Causes of events
### Types of causes

<table>
<thead>
<tr>
<th>Internal causes</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caused by unusual use</td>
<td>Over operational limits like &quot;hard landing&quot;, &quot;over-g maneuver&quot;</td>
</tr>
<tr>
<td>Caused by internal dysfunction</td>
<td>Excessive heat</td>
</tr>
<tr>
<td></td>
<td>Excessive pressure</td>
</tr>
<tr>
<td></td>
<td>Excessive vibration</td>
</tr>
</tbody>
</table>

**Step 2**: Identify each usage phase of product (eg operation, maintenance, storage, transport). If necessary each one may be subdivided into sub phases (eg for an aircraft: take off, flight and landing).

**Step 3**: Apply pertinent type of causes to the different phases of product life to identify all possible events.

Examples of special events due to external source:
- Impact on external skin due to ground handling equipment during servicing
- Impact on external skin due to cargo handling equipment during transportation preparation
- Multiple impacts due to hail
- Impacts on inner external skin due to runway debris during take off
- Lightning strike

Examples of special events due to internal source:
- Water entrapment due to a leak
- Excessive stress on a tank structure due to dysfunction of pressure relief valve during a refueling operation on ground
- Excessive temperature in a bay in case of warm air leak
- An unexpected wear-out of mechanical parts

**Step 4**: Analyze each possible event in order to characterize its probability of occurrence.

For some special events a quantitative approach may be applied based on feedback from previous similar products. For example: number of bird strikes happening every year.

When it is not possible to get sufficient statistical elements, a qualitative approach may be applied using **Table 4**.

**Table 4** Probability of occurrence of special events

<table>
<thead>
<tr>
<th>Rating</th>
<th>Occurrence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Extremely unlikely</td>
<td>A special event whose probability of occurrence is essentially zero</td>
</tr>
<tr>
<td>2</td>
<td>Remote likelihood</td>
<td>A special event whose probability of occurrence is unlikely. Rare numbers of special events are likely to happen.</td>
</tr>
<tr>
<td>3</td>
<td>Occasional</td>
<td>A special event whose probability of occurrence is occasional. A few special events are likely to happen.</td>
</tr>
<tr>
<td>4</td>
<td>Reasonably probable</td>
<td>A special event whose probability of occurrence is moderate. A certain number of special events are likely to happen.</td>
</tr>
</tbody>
</table>
Step 5: Depending on the occurrence rating from the previous step, categorize the special event as maintenance significant or not for continuation in the process (refer to Para 3.3). The rating level may be used to determine a threshold to tailor analysis activities.

3.2 Identify potential damage to be considered

This sub process is based on a technology approach. Experience shows that technologies used can be more or less sensitive to damage. The approach consists of identifying the technologies used on the different subsystems/items of the product and analyzing them through two considerations:

- Is the technology well-known or new?
- Is the technology damage proof or sensitive to damage?

Identify the potential damage to be considered using the following methodology:

Step 1: For all subsystems/items of the product, identify the technologies used and characterize it regarding knowledge of its behavior.

Table 5 may be used for a qualitative approach to characterize this knowledge. The grade of accuracy of the rating may be adjusted depending on the individual program.

Table 5 Technology behavior knowledge rating

<table>
<thead>
<tr>
<th>Rating</th>
<th>Technology behavior knowledge</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Well known</td>
<td>This technology has been used on many similar programs</td>
</tr>
<tr>
<td>2</td>
<td>Known</td>
<td>This technology has been used on similar programs but has been marginally modified for the intended program</td>
</tr>
<tr>
<td>3</td>
<td>Quite new</td>
<td>This technology has been already used on similar recent program but very little feedback is available</td>
</tr>
</tbody>
</table>

Step 2: Evaluate sensitivity of the technology.

For each technology identified in Step 1, identify the possible damage types that may occur. For example:

- Corrosion on metallic parts (general, galvanic or other types of corrosion)
- Stress corrosion on a metallic assembly with constraints installed
- Delaminating or humidity absorption on composite material

Evaluate its sensitivity to damage types in relation with possible damage sources.

Table 6 can be used for a qualitative approach to characterize this sensitivity. The grade of accuracy of the rating may be adjusted depending on the individual program.
### Table 6  Technology sensitivity rating

<table>
<thead>
<tr>
<th>Rating</th>
<th>Sensitivity degree</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Extremely low</td>
<td>This technology has an extremely low chance of damage during product life</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>This technology has a very low chance of damage during product life</td>
</tr>
<tr>
<td>3</td>
<td>Medium</td>
<td>This technology has a low chance of damage during product life</td>
</tr>
<tr>
<td>4</td>
<td>High</td>
<td>This technology is likely to be damaged during product life</td>
</tr>
<tr>
<td>5</td>
<td>Extremely high</td>
<td>This technology is very likely to be damaged during product life</td>
</tr>
</tbody>
</table>

**Step 3**: Select the association technology/potential damage as relevant or not to go ahead in the process. Combining the technology behavior knowledge rating (refer to Table 5) and the technology sensitivity rating (refer to Table 6) from the previous steps can result in Fig 2 example. It can be used to tailor the process according to requirements. The selection threshold can be adjusted depending on the needs of the individual program.

![Technology/damage evaluation rating](ICN-B6865-S3000L0088-001-01)

**3.3 Define element or part of the product concerned**

The link between the two approaches (special event and technology) can be done in terms of the resulting potentially damageable items:

- A relevant special event can generate damage to an item that is made with sensitive technology
- An element made with sensitive technology may be damaged because of a special event

The two approaches are complementary and identify potentially damageable items that may be encountered during product life. To identify the potentially damageable items to be considered, the following methodology is provided:

**Step 1**: For each special event selected, identify the affected items of the breakdown structure.

**Step 2**: For each association technology/damage selected, identify the affected items of the breakdown structure.

**Step 3**: For each item of the breakdown structure identified at the previous steps, analyze the installation area/design to evaluate exposure to the different threats. For example, level of exposure to corrosion depends on where the structural part is located. An item in a landing gear bay is very exposed to corrosion while a watertight bay very rarely opened is significantly less exposed to corrosion.
Step 4: Using the results of the previous steps, decide for each potentially damageable items whether or not it should be considered in the process.

3.4 Criticality analysis
In order to focus on the most significant special events/damages, a criticality analysis should be considered.

For each potentially damageable item evaluate the criticality in terms of consequence on the product service life.

The following aspects should be taken into consideration:

− Does the damage result in lowering safety level?
− Does the damage result in lowering availability?
− Does the damage result in significant ownership deterioration?
− Does the damage result in significant cosmetic deterioration?

3.5 Maintenance task requirement identification
Previous steps of the process have identified and selected the significant item of the breakdown structure affected by a special event or a possibility of being damaged during service life.

Each significant item and event must be analyzed in terms of the maintenance requirement.

This involves identifying the different maintenance tasks required to be able to:

− Detect, localize, measure or follow each damage that might occur (e.g., visual inspection, non-destructive test)
− Restore the function of the element (e.g., remove and replace task, repair task, refurbish task)

Some elements can be grouped into families and associated damages are candidates for standard repair procedures.

4 Inputs
4.1 Physical breakdown
Physical breakdown (or product breakdown) provides an indented decomposition of the product into systems, subsystems, and items.

4.2 Product usage description sheet
Product usage description sheet provides information like operation environment and maintenance situation.

4.3 Statistical elements on special events
Feedback from previous similar programs to quantify the occurrences of special events.

4.4 Product definition
Elements of the product definition needed are those regarding technologies employed in the different products, systems, subsystems, and items.

5 Outputs
5.1 Special event and damage analysis tabular report
It is recommended that a tabular report be developed that provides all important information and outputs in a convenient form. These results should be recorded in an LSA database in order to keep the traceability of maintenance task requirements.
An example of a damage and special event analysis tabular report is given in Fig 3. A description of each column is contained in Table 7.

Note
Although not shown in Fig 3, header information to include release date, issue number, author’s name, approval signatures, information required for identification of the maintenance task and other program identifiers should be included.

This tabular report should be tailored to meet program needs.

<table>
<thead>
<tr>
<th>Column Description</th>
<th>Significant element identification</th>
<th>Special event description</th>
<th>Special event occurrence rating</th>
<th>Technology description</th>
<th>Damage description</th>
<th>Technology/damage evaluation rating</th>
<th>Exposure evaluation rating</th>
<th>Criticality analysis result</th>
<th>Maintenance task requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Significant element identification</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element name. This element name can be followed by its breakdown element identifier. Some elements (typically structural elements) can be grouped into families in order to minimize the number of entries in the table.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>Special event description</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The special event is described by a complete name. The special event may affect more than one element of the breakdown structure. In this case identify each one on a different line.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>Special event occurrence rating</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gives the rating and description of occurrence.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>Technology description</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Give a short technical description of the technology. The technology may concern more than one element of the breakdown structure, in this case try to group them by families in order to minimize the number of entries.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>Damage description</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Give a short description of damage. If more than one damage can affect the same element then identify each one on a different line.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 6      | **Technology/damage evaluation rating**  
Give the rating and description explaining the rating in terms of sensitivity and newness. |
| 7      | **Exposure evaluation description**  
Give a short technical description of the location of the element identified and how it is exposed to threats. |
| 8      | **Criticality analysis result**  
Describe the justification of importance of damage regarding its consequences. |
| 9      | **Maintenance task requirement**  
A short summary or title including the type of maintenance task and the element on which it applied. If more than one maintenance task is needed to cover the same element then identify each one on a different line. Conversely, the same maintenance task can address more than one element or damage. This is typically the case of a standard repair procedure. |

### 5.2 Complementary analysis

#### 5.2.1 Influence on design

Early in the program development/design process there is the opportunity to influence the design from the standpoint of special events and damage occurrences. Design modifications can reduce/eliminate special events and damages, for example:

- Adding protection
- Enhancing accessibility/removals
- Changing the technology

Such possible influence on design should be appropriately managed and milestone in the design process.

### 5.3 Lessons learned

This section could give examples of best practices or bad experiences to be taken into account for future studies.
Chapter 9

*Logistics related operations analysis*

Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistics related operations analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>2 Logistics relevant operations</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Usage supporting aspects</td>
<td>3</td>
</tr>
<tr>
<td>2.1.1 Preparation for usage</td>
<td>3</td>
</tr>
<tr>
<td>2.1.2 Servicing</td>
<td>3</td>
</tr>
<tr>
<td>2.1.3 Adjusting</td>
<td>4</td>
</tr>
<tr>
<td>2.1.4 Weighing</td>
<td>4</td>
</tr>
<tr>
<td>2.1.5 Loading and unloading</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Packing, handling, storage and transport aspects</td>
<td>5</td>
</tr>
<tr>
<td>2.2.1 Packing and unpacking</td>
<td>5</td>
</tr>
<tr>
<td>2.2.2 Handling</td>
<td>5</td>
</tr>
<tr>
<td>2.2.3 Storage</td>
<td>6</td>
</tr>
<tr>
<td>2.2.4 Stacking</td>
<td>6</td>
</tr>
<tr>
<td>2.2.5 Lifting</td>
<td>6</td>
</tr>
<tr>
<td>2.2.6 Transportation</td>
<td>6</td>
</tr>
<tr>
<td>2.2.7 Mooring</td>
<td>7</td>
</tr>
<tr>
<td>2.2.8 Shoring</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Software and data aspects</td>
<td>7</td>
</tr>
<tr>
<td>2.4 Product or system recovery</td>
<td>7</td>
</tr>
<tr>
<td>2.5 Special scheduled maintenance</td>
<td>7</td>
</tr>
<tr>
<td>2.6 Disposal and recycling</td>
<td>8</td>
</tr>
<tr>
<td>2.7 Logistics relevant operations out of the ordinary</td>
<td>8</td>
</tr>
<tr>
<td>3 Documentation of logistics relevant operations tasks</td>
<td>8</td>
</tr>
<tr>
<td>3.1 LSA database aspects</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Operational and customer requirements as a source</td>
<td>8</td>
</tr>
<tr>
<td>4 Checklist of logistics relevant operations tasks</td>
<td>8</td>
</tr>
</tbody>
</table>

List of tables

1 References ......................................................................................................................2
2 Checklist for logistics related operations analysis ......................................................8

List of figures

1 Logistics related operations analysis within technical documentation ....................2
2 Typical servicing tasks ................................................................................................4
3 Typical PHST tasks ........................................................................................................5
1 General

1.1 Introduction

Beside the activities concerning maintenance and repair of a product, there are additional aspects concerning the operation and the handling to be considered. Logistic relevant operations are tasks, which cannot be assigned to an area of direct usage of a product (documented in operating instructions) or an area of maintenance (documented in maintenance manual). However, these tasks can be of importance for the proper usage of any product. Many aspects such as ease of operation, usability, flexibility of usage or mobility are restricted by logistics relevant operations tasks. The border between pure usage of a product and the logistics relevant operations is subjective and it must be decided individually for each project, how and where these aspects will be documented. Sometimes it can be difficult to assign a maintenance task exactly to one of these two areas.

It is recommended to clarify at an early stage of any project the responsibility of both, performance of logistic analysis and documentation of logistics relevant operations.

![Diagram of technical documentation]

**Fig 1** Logistics related operations analysis within technical documentation

All tasks that are identified by the logistics relevant operations analysis will close the possible gap between user manual and maintenance manual.
1.2 Objective
This chapter should be a guideline for identifying relevant operational activities. To support the analyst, a detailed list of potential operational or handling activities is provided in Para 4.

1.3 Scope
This chapter is directed to the logistics personnel responsible for analyzing the logistics relevant operational tasks.

2 Logistics relevant operations
In the following sections the typical logistics relevant operations are described in more detail to explain the concept. The collection of examples is limited to the most common tasks. There can be more activities, especially in relation to environmental conditions (eg preparation for transport under extreme climatic conditions such as an arctic environment). For each example a short description is given along with special aspects to be considered.

The identification of logistics relevant operations, including the requirements concerning personnel, support equipment, consumables, spare parts, facilities and required training, is an important area of logistics analysis tasks. Some of the tasks listed below require very early consideration in the development life cycle and some can be considered later, eg when a prototype of the Item under Analysis (IuA) is available.

2.1 Usage supporting aspects
The activities that support the usage of the IuA are described in the following paragraphs.

2.1.1 Preparation for usage
This aspect describes the tasks required to prepare a product for a special operation. This preparation can include, for example, the exchange of special equipment within the IuA to provide a special capability. In addition to the modification of the IuA by changing some pieces of equipment, it can be possible that peripheral products/components must be prepared for a proper usage.

Typical preparation tasks can be classified as servicing tasks, (eg the cleaning of windows before driving a vehicle). Others are typical preparation tasks, which describe the complete conversion of a product for another use case, such as the change of the tooling within a machine to produce a new product line. Typical examples for tasks concerning preparation for usage, which can be considered by Logistics Relevant Operations Analysis, are the following:

− Preparation of machines for new production segment (eg by exchange of tooling or dies)
− Role change of a vehicle (role change of a normal transport vehicle to an ambulance transport vehicle)
− Preparation of an aircraft for a reconnaissance mission by installing special equipment
− Preparation of a ship including each necessary equipment for sailing

2.1.2 Servicing
The term servicing can describe a wide range of tasks performed on a product (also in connection with usage preparation). Other aspects are handling after usage or the upkeep of the system. This includes, for example, a simple visual inspection for foreign objects during washing or cleaning procedures as well as simple visual inspections concerning damage or the general condition of equipment (eg condition of tires).
Servicing tasks should be examined by the means of a Maintenance Task Analysis (MTA) to identify the requirements. Servicing tasks can have an important impact on resources, to include personnel, support equipment (eg for lubrication), consumables and facilities (eg washing facilities with oil separator and water recycling equipment). Typical examples for tasks in connection with servicing are the following:

- Engine washing of any vehicle
- Lubrication of a gear
- Oil change of an engine
- Polishing and following preservation of a vehicle
- Bleeding of brakes
- Change of hydraulic liquid
- Replenishment of fluids
- Desalination and rinsing of equipment after diving

### Adjusting

The typical adjust tasks can also be performed in connection with preparation for usage. Product functionality is not changed, but precision and quality considerations are addressed as preconditions for the usage of the product. Typical examples for tasks in connection with adjusting are the following:

- Leveling of the entire product as a basic task
- Calibration of measuring instruments
- Adjustment of the sight of a gun

### Weighing

The analysis of weighing tasks involves the preparation of the IuA for weighing and the procedure itself. This includes information on the weighing equipment to be used. The purpose of the weighing procedure should be defined and agreed to between the contractor and the customer (eg weighing as a mission preparation task or a task to establish proof of specification requirements of the product).

### Loading and unloading

Loading and unloading procedures should be analyzed for each product that can be used for the transport of cargo. This information should be collected in an early stage in the development life cycle. Examples of loading and offloading techniques, interior layout, floor loadings, location and strength of lashing points, methods of stowing and securing, capacities and dimensions of compartment and doors should be documented carefully. The following aspects should be considered:

- What kind of cargo is planned to be transported?
- Which size and weight parameters must be considered?
− Is the expected cargo sensitive to special impacts (acceleration, magnetic or electric fields, pushes, humidity, etc)?
− Is the cargo critical or even dangerous in case of improper handling?
− What type of cargo securing (lashing and lashing points, stowing) is required?
− Will a container concept be employed?
− Are special loading devices required because of the type of the load (eg rolling devices)

The requirements for special support equipment for loading and unloading should also be addressed.

− Is it necessary to use special support equipment or vehicles for the loading/unloading?
− Is special support equipment required to secure the cargo?

### 2.2 Packing, handling, storage and transport aspects

This paragraph describes the activities in connection with the packing, handling, storage and transport (PHST) of the IuA during other than normal use.

#### 2.2.1 Packing and unpacking

A packing concept for the IuA and/or for components should address the following:

− Is packing required for short term storage and/or for long term storage?
− Is packing required for transportation and what kind of transportation will be carried out?
− Is a special container required for storage or transport?
− Is special preservation required because of extreme climatic conditions during the storage or transportation period (eg sea transport)?
− Is it required to unpack and repackage the IuA during transportation or storage period, for example, to perform maintenance activities (storage tasks)?
− What is required for the unpacking and removal of preservation concerning support equipment and facilities?

#### 2.2.2 Handling

Handling tasks for the IuA should address the following:

− Safety precautions and limitations due to handling
− Instructions necessary to park, tow, winch or move the IuA in any way, other than normal usage.
− Instructions necessary to jack the IuA including jacking points, required adapters, supports for special components, balance weights, jacking procedures, etc
− Additional equipment and materials required when handling the IuA (eg tow bars or cables).
2.2.3 Storage

To guarantee product functionality during a storage period and at the conclusion of storage, storage procedures should be analyzed and documented. Storage tasks should address the following:

− Is the required storage considered long term storage or short term storage solution?
− Is the product/component to be stored of special sensitivity?
− Is it necessary to remove components from the product to be stored and to store these components separately under special conditions?
− What type of inspection and preventive maintenance is required to safeguard structural and system integrity during storage (e.g., wheel rotation, provision with electrical power, engine running, pressure checks)? Provide a timescale for such in-storage maintenance.
− Are there any extreme climatic conditions which are expected during storage period (heat, frost, humidity)?
− Are there any special techniques required for entry into storage (e.g., cleaning and preservation, fluid system draining/replenishing, static grounding, protective blanking, removal of special components)?
− Are there any special techniques required for removal from storage and to return to usage (e.g., cleaning and removal of preservation, fluid system replenishing, re-installation of special components, functional checks, preparation for usage)?
− What kind of securing is required during storage period (e.g., mooring, blocking of movable components, securing against light in a dark storage place)?
− Is it necessary to consider the storage time in connection with the life of the stored product/component?

2.2.4 Stacking

A special aspect of storage and transportation of any product/component is stacking. What are the safety requirements for stacking for simple storage or for transport? For example, the impact of external events such as pushing should especially be analyzed for storage.

2.2.5 Lifting

A lifting concept should include all necessary procedures to lift the IuA with corresponding hoist devices, cranes, jacks or slings for the following reasons:

− Lifting for transportation or loading purposes
− Lifting for repair or maintenance purposes
− Lifting for recovery

In addition to the lifting of the complete product, the lifting of components should be considered as well (e.g., the lifting of the engine of an aircraft).

2.2.6 Transportation

Based on customer requirements, the transportability of the product should address the following:

− What is the effort and duration of the preparation for transport and for recovery after transport?
− What are the requirements for preparation for transport and for recovery after transport concerning personnel, tools, consumables, and facilities?
− What are the additional environmental requirements for transportation under special conditions (e.g., extreme climatic conditions such as sea transport, sandy desert environment, humidity, heat, frost)?
− What are the additional safety requirements for transportation under special conditions (e.g., mooring within a transport aircraft in case of air transport)?
− What are the required servicing actions during the transportation period (especially during long journeys)?
Is it necessary to remove components for transportation or to disassemble the entire IuA down to a specific level?
Should a container concept be considered? Is the usage of sledges and/or pallets acceptable?

2.2.7 Mooring
Analysis of mooring tasks for the IuA must be addressed under all weather conditions or for transportation within any transportation vehicle. Mooring is considered as a long or short term activity, whose purpose is to tie down or otherwise secure the IuA to the ground or within a transportation vehicle to avoid any damage. Also, information such as special techniques applicable to ballasting, definition of lashing points and installation/use of special support equipment applicable to mooring should be included.

2.2.8 Shoring
Similar to mooring, shoring should be analyzed concerning shoring points, shoring procedures and the shoring equipment used during, for example, maintenance, repair and recovery.

2.3 Software and data aspects
Logistics relevant operations related to software are addressed in Chap 13. Software Support Analysis.

2.4 Product or system recovery
Analysis concerning recovery should include any information on planned recovery procedures and the required support equipment needed to recover any IuA from any condition to which it may be subjected.

Product recovery during the testing phase should be addressed separately. It is important to guarantee a fast reaction to any unexpected event, where it is necessary to recover the product from an undesired condition. It must be clear, what action is required to avoid any endangering of personnel or environment. A recovery plan for all emergency situations should be detailed enough to avoid endangering personnel or the environment.

Recovery also includes rescue procedures or safeguarding of an area after a catastrophic event. All procedures that are necessary to rescue any product after an accident must be analyzed before the first usage of the product. For example, an aircraft performing its first flight marks a specific risk of a crash. All required activities after a potential crash must be analyzed and documented carefully before the first usage of the system. Recovery training should be included as part of the analysis process in order to reduce the risk of extended damage to humans or the environment in case of a catastrophic event. All aspects must be covered, eg an accident on land, in water (eg the foundering of a ship with the requirement to salvage the wreckage) or within areas that are difficult to access.

2.5 Special scheduled maintenance
In general, the identification of Scheduled Maintenance is described in Chap 10. However, servicing tasks or tasks in connection with preparation for usage can be of a scheduled nature, but are not identified by the extended Scheduled Maintenance Analysis (SMA). These tasks can be of high impact, eg for personnel requirements, because they are performed frequently (eg even daily). Examples include the following:
- Visual check before the start of usage (eg starting of an engine)
- Visual check after special missions
- Scheduled replenishment of fuel, fluids or other consumables to guarantee proper function
- Cleaning before or after each usage
2.6 Disposal and recycling
Analysis concerning requirements after the life cycle of a product is of increasing importance. This includes both the disposal of the complete product, and the handling of components and consumables to be disposed of during the life cycle of the product. For this reason, these aspects are described in detail in Chap 17 - Disposal.

2.7 Logistics relevant operations out of the ordinary
Other logistics relevant operations include, but are not limited to the following and should be addressed in the analysis process:
− Decontamination of a vehicle
− Disinfection of personnel before starting a job
− De-icing of an aircraft or ship
− Check of electrical charge
− Check of the strength of magnetic fields during storage
− Paperwork for statistic purpose

3 Documentation of logistics relevant operations tasks

3.1 LSA database aspects
The identification of any required logistics relevant operation should be documented within the LSA database using the appropriate information code. It is recommended that logistics relevant operations tasks be documented at the product level or the appropriate breakdown level (use of a mixed physical/functional breakdown provides the most flexibility). This will support the exchange of this data with for example, S1000D for technical publications.

3.2 Operational and customer requirements as a source
In general, the requirements for logistics relevant operations tasks can be derived from the operational and customer's requirements, which are collected and documented within the establishment of product usage data. This process is described in detail in Chap 3. It is recommended to use the content of the relevant documents ORD (Operational Requirements Document) and CRD (Customer Requirements Document) as a first input for the Logistics Relevant Operations Analysis. Each aspect concerning usage of the IuA, which can be found in the ORD or CRD, can trigger the requirement for a logistics relevant operations task.

4 Checklist of logistics relevant operations tasks
The following alphabetical checklist for potential relevant activities is provided as a guide.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Analysis recommended at which phase</th>
<th>Probable large impact on requirement for facilities</th>
<th>Probable large impact on requirement for special support equipment</th>
<th>Probable large impact on design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjusting</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Bleeding</td>
<td>When necessary</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Calibration</td>
<td>When necessary</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Checking</td>
<td>When necessary</td>
<td>No</td>
<td>No</td>
<td>Possible</td>
</tr>
<tr>
<td>Cleaning</td>
<td>Early phase</td>
<td>Yes</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Conservation</td>
<td>Early phase</td>
<td>Yes</td>
<td>Possible</td>
<td>No</td>
</tr>
</tbody>
</table>

Applicable to: All
<table>
<thead>
<tr>
<th>Activity</th>
<th>Analysis recommended at which phase</th>
<th>Probable large impact on requirement for facilities</th>
<th>Probable large impact on requirement for special support equipment</th>
<th>Probable large impact on design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container concept</td>
<td>When necessary</td>
<td>No</td>
<td>No</td>
<td>Possible</td>
</tr>
<tr>
<td>Conservation removal</td>
<td>Early phase</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Desalination</td>
<td>When necessary</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Disposal</td>
<td>Early phase</td>
<td>Possible</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Draining</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Jacking</td>
<td>Early phase</td>
<td>Possible</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Leveling</td>
<td>Early phase</td>
<td>Possible</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Lifting</td>
<td>Early phase</td>
<td>Possible</td>
<td>Yes</td>
<td>Possible</td>
</tr>
<tr>
<td>Loading cargo</td>
<td>Early phase</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Loading data</td>
<td>Early phase</td>
<td>Yes</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Loading Software</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Lubrication</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Mooring</td>
<td>When necessary</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Oil Change</td>
<td>When necessary</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Packing</td>
<td>When necessary</td>
<td>Possible</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Parking</td>
<td>When necessary</td>
<td>Possible</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Polishing</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Preservation</td>
<td>Early phase</td>
<td>Possible</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Recovery</td>
<td>Early phase</td>
<td>No</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Recycling</td>
<td>When necessary</td>
<td>Possible</td>
<td>Possible</td>
<td>Yes</td>
</tr>
<tr>
<td>Replenishment</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Rinsing</td>
<td>When necessary</td>
<td>Possible</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Safety precautions</td>
<td>Early phase</td>
<td>Possible</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Securing</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Shoring</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Stacking</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Storage</td>
<td>Early phase</td>
<td>Yes</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Activity</td>
<td>Analysis recommended at which phase</td>
<td>Probable large impact on requirement for facilities</td>
<td>Probable large impact on requirement for special support equipment</td>
<td>Probable large impact on design</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------</td>
<td>----------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Towing</td>
<td>When necessary</td>
<td>No</td>
<td>Yes</td>
<td>Possible</td>
</tr>
<tr>
<td>Transportation</td>
<td>Early phase</td>
<td>Possible</td>
<td>Yes</td>
<td>Possible</td>
</tr>
<tr>
<td>Unloading cargo</td>
<td>Early phase</td>
<td>Possible</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Unloading data</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Unloading Software</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Unpacking</td>
<td>Early phase</td>
<td>Possible</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Upkeep</td>
<td>Early phase</td>
<td>Possible</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Washing</td>
<td>Early phase</td>
<td>Possible</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>Weighing</td>
<td>Early phase</td>
<td>Possible</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Winching</td>
<td>When necessary</td>
<td>No</td>
<td>Possible</td>
<td>No</td>
</tr>
</tbody>
</table>
Chapter 10

Scheduled maintenance analysis

Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduled maintenance analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Terms, abbreviations and acronyms</td>
<td>2</td>
</tr>
<tr>
<td>2 Division of scheduled maintenance analysis</td>
<td>3</td>
</tr>
<tr>
<td>2.1 System and power plant analysis</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Structure analysis</td>
<td>3</td>
</tr>
<tr>
<td>2.3 Zonal analysis</td>
<td>4</td>
</tr>
<tr>
<td>3 Preventive or scheduled</td>
<td>4</td>
</tr>
<tr>
<td>4 LSA candidates and SMA candidates</td>
<td>5</td>
</tr>
<tr>
<td>5 Scheduled maintenance analysis procedure</td>
<td>5</td>
</tr>
<tr>
<td>6 Thresholds and intervals from scheduled maintenance analysis</td>
<td>6</td>
</tr>
<tr>
<td>6.1 PO threshold</td>
<td>6</td>
</tr>
<tr>
<td>6.2 PE threshold or interval</td>
<td>6</td>
</tr>
<tr>
<td>6.3 Mixture of PO and PE thresholds</td>
<td>6</td>
</tr>
<tr>
<td>6.4 Triggers</td>
<td>7</td>
</tr>
<tr>
<td>7 Thresholds and intervals from other sources</td>
<td>7</td>
</tr>
<tr>
<td>8 LSA tasks</td>
<td>8</td>
</tr>
<tr>
<td>9 Scheduled maintenance harmonization and task packaging process</td>
<td>8</td>
</tr>
</tbody>
</table>

List of tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>Terms, abbreviations and acronyms</td>
<td>3</td>
</tr>
<tr>
<td>Terminology - preventive and scheduled maintenance</td>
<td>4</td>
</tr>
</tbody>
</table>

List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preventive/scheduled maintenance</td>
<td>5</td>
</tr>
<tr>
<td>PO threshold</td>
<td>6</td>
</tr>
<tr>
<td>PE interval</td>
<td>6</td>
</tr>
<tr>
<td>PO and PE thresholds dependent from each other</td>
<td>7</td>
</tr>
<tr>
<td>Triggers</td>
<td>7</td>
</tr>
<tr>
<td>Main aspects for decrease and increase of intervals/ thresholds (example)</td>
<td>8</td>
</tr>
<tr>
<td>Packages from single scheduled LSA tasks</td>
<td>9</td>
</tr>
</tbody>
</table>
References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 21</td>
<td>Terms, abbreviations and acronyms</td>
</tr>
<tr>
<td>MSG-3</td>
<td>ATA MSG 3 - Operator/Manufacturer Scheduled Maintenance Development</td>
</tr>
<tr>
<td>RCM</td>
<td>Eg MIL-STD-1843 (USAF) - Reliability-centered maintenance for aircraft, engines and equipment</td>
</tr>
<tr>
<td></td>
<td>Eg MIL-STD-2173 (AS) - Reliability-centered maintenance for naval aircraft, weapon systems and support equipment</td>
</tr>
<tr>
<td></td>
<td>UK MoD DefStan 02-45 - Requirements for the application of reliability-centred maintenance - techniques to HM ships, submarines, royal fleet auxiliaries and other naval auxiliary vessels</td>
</tr>
<tr>
<td>S4000M</td>
<td>International procedure handbook for the development of scheduled maintenance programs for military aircrafts</td>
</tr>
</tbody>
</table>

1 General

1.1 Introduction

The identification of required scheduled maintenance is of vital importance for the operation of a complex product. Safety, economic, environmental and ecological aspects must be considered very carefully. It is a common tendency to minimize scheduled maintenance and to come to condition based maintenance concepts. However, experience (especially in safety critical areas) shows that a certain amount of scheduled maintenance can not be avoided.

1.2 Objective

This chapter introduces the well known methods of Scheduled Maintenance Analysis (SMA), like S4000M, MSG-3 or Reliability Centered Maintenance (RCM) analysis, and how they should be linked to the LSA process and where the connecting aspects are.

1.3 Scope

This chapter is directed at logistics personnel who perform SMA and look for an appropriate place to document the results of this extensive analysis. The LSA database is the correct place, not only to document the results from the analysis which provides the identification of a scheduled maintenance activity, but also to document the results of the corresponding maintenance task analysis.

It should be noted that the extensive analysis activities are carried out outside the LSA database (eg FMECA, LORA, SMA). However, the results of these technical/logistic analysis activities have to be documented in the LSA database as agreed between contractor and customer for a specific project.

1.4 Terms, abbreviations and acronyms

The following specific terms, abbreviations and acronyms used within this chapter. More special terms are explained directly within the appropriate paragraphs. Refer to Chap 21 for a full listing of Terms, abbreviations and acronyms.
### Table 2 Terms, abbreviations and acronyms

<table>
<thead>
<tr>
<th>Abbreviation/Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preventive maintenance</td>
<td>Maintenance activities that prevent the occurrence of critical failures or damages in conjunction with safety, economical or ecological aspects. Preventive maintenance also includes activities following special events (for these events chronological intervals or other regular threshold cannot be defined)</td>
</tr>
<tr>
<td>Scheduled maintenance</td>
<td>Maintenance activities that prevent the occurrence of critical failures or damages in conjunction with safety, economical or ecological aspects. These maintenance tasks are defined with a corresponding interval or threshold (eg after certain time, cycles, rounds, distance). Scheduled maintenance is a subset of the preventive maintenance.</td>
</tr>
<tr>
<td>SMA candidate</td>
<td>Element of the breakdown, which is identified as a potential candidate for which a scheduled or preventive maintenance task could be necessary.</td>
</tr>
</tbody>
</table>

## 2 Division of scheduled maintenance analysis

The Scheduled Maintenance Analysis (SMA) documented in international specifications such as S4000M, MSG-3 (civil specification) or RCM (Reliability Centered Maintenance, military specification) are grouped into three general main areas. These are:

- System and power plant analysis
- Structural analysis
- Zonal analysis

Each of these analysis methods has its own methodology and own procedures to determine scheduled/preventive maintenance tasks and the corresponding intervals.

### 2.1 System and power plant analysis

The method for determining the scheduled maintenance tasks and intervals for systems and power plants, including components and equipment, uses a progressive logic diagram described in the various SMA specifications. This logic is the basis for an evaluation technique applied to each maintenance significant item (eg system, subsystem, module, component, accessory, unit, part), using the technical data available. In principle, the evaluations are based on the item's functional failures and the respective failure causes. As additional knowledge and experience of the product is gathered during the inservice phase (eg fault/damage reports, maintenance work orders), the existing tasks shall be reviewed periodically.

### 2.2 Structure analysis

The structure analysis contains guidelines for developing applicable and effective scheduled maintenance tasks with intervals for the mechanical structure of a product (eg vehicles, aircrafts, ships, tanks). These are designed to relate the scheduled maintenance tasks to the consequences of structural failure or damage that has remained undetected.

Each structural item with details (if selected) is assessed in terms of its significance to continued operational safety, impact on availability for operation or missions, susceptibility to any form of failure or damage, and the degree of difficulty involved in detecting such failure or damage.

Once this is established, scheduled structural maintenance can be developed to be effective in detecting and preventing structural degradation due to fatigue failure, environmental deterioration and/or accidental damage. Resulting scheduled tasks are to be performed from the beginning of the inservice phase throughout the operational life of the product under analysis. As additional knowledge and experience of the product is gathered during the
in-service phase (e.g., fault/damage reports, maintenance work orders), the existing tasks shall be reviewed periodically.

2.3 **Zonal analysis**

Zonal inspections for products may be developed from application of a zonal analysis Procedure coming from the various SMA specifications. This requires a summary review of each zone of the product and normally occurs as analyses of structures, systems and power plants are being concluded. Identified general visual inspection tasks from those analysis steps may subsequently be included in the zonal inspections where intervals can be harmonized with the zonal inspection intervals. As additional knowledge and experience of the product is gathered during the in-service phase, the existing tasks shall be reviewed periodically.

3 **Preventive or scheduled**

For a common understanding of terminology, the terms *preventive* and *scheduled* should be defined clearly:

<table>
<thead>
<tr>
<th><strong>Term</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Preventive</td>
<td>All activities to avoid critical failures or other critical situations should be summarized as preventive activities. This includes:</td>
</tr>
<tr>
<td></td>
<td>− Scheduled maintenance activities with a replace, repair or overhaul activity such as the replacement of components after a specified period of use (e.g., time change items) or after another specific interval (e.g., rounds, cycles, distance)</td>
</tr>
<tr>
<td></td>
<td>− Scheduled inspections after specific intervals or thresholds (tests, checks or similar activities to make a critical failure obvious)</td>
</tr>
<tr>
<td></td>
<td>− Activities after damages or special events to establish the condition of the analyzed technical system after an event beyond of normal usage</td>
</tr>
<tr>
<td>Scheduled</td>
<td>All activities to avoid critical failures or other critical situations with a certain interval or threshold</td>
</tr>
<tr>
<td></td>
<td>− Scheduled maintenance activities with a replace, repair or overhaul activity such as the replacement of components after a specified period of use (e.g., time change items) or after another specific interval (e.g., rounds, cycles, distance)</td>
</tr>
<tr>
<td></td>
<td>− Scheduled inspections after specific intervals or thresholds (e.g., tests, checks or similar activities to make a critical failure obvious)</td>
</tr>
</tbody>
</table>

Corresponding to this definition given in Table 3, scheduled maintenance can be regarded as a subset of preventive maintenance as shown in Fig 1. The main characteristic of a scheduled maintenance task is the existence of a specified interval or threshold.
Preventive maintenance

Scheduled maintenance

- Scheduled activities (e.g., inspection, lubrication, test, replacement of time change items) after time limitations, like time intervals or thresholds
- Scheduled activities (e.g., inspection, lubrication, test, replacement of time change items) after non-time limitations like intervals and thresholds based on e.g., cycles, rounds, distances
- Inspection activities after special events

Fig 1 Preventive/scheduled maintenance

4 LSA candidates and SMA candidates

The criteria for the selection of SMA candidates and LSA candidates are different. The selection rules documented in the various SMA specifications are mainly safety driven. In newer specifications such as S4000M, economic and ecological aspects are considered. The selection rules of typical LSA candidates are mainly driven by maintenance relevance. Nevertheless, there is an interconnection between these two types of candidates.

A Maintenance Significant Item (MSI) selected for SMA generally becomes a LSA candidate, if scheduled maintenance activities are identified for the item. These identified tasks and their intervals will be documented and further analyzed for task requirements within the LSA database.

For structural items, it can be necessary to expand the system breakdown for additional items representing only special areas or details of a large structural item (e.g., an aircraft wing structure), for which a structure analysis must be performed. These artificial breakdown elements are only introduced in order to allow the assignment of the analysis results to a corresponding object. This method is described in detail in S4000M, structure analysis. The corresponding scheduled or preventive tasks and their intervals, identified within the SMA in the case of structure, will be documented and further analyzed for the consequent task requirements within the LSA database against a structural breakdown element, which then becomes a LSA candidate.

The zonal analysis is a special area where the item of SMA is not a physical hardware component or a group of hardware components, but a local area within a product, where a lot of different items of different types (including the carrying structure) may be located. It is possible that the zone may only contain structural items. All identified scheduled maintenance tasks (normally general visual inspections) and their intervals will be documented and further analyzed for task requirements consequently within the LSA database. The method for addressing LSA candidates for zones must be defined carefully and harmonized with the customer during the LSA guidance conference. It is recommended that the SMA results for zones under dedicated breakdown elements are gathered and the required maintenance tasks be grouped by interval criteria.

5 Scheduled maintenance analysis procedure

The SMA is driven by three main aspects. First of all, safety related aspects are the most important ones. Economic and ecological aspects are also of increasing importance, especially the ecological effects.

Detailed analysis of potential failures and damage, their effects and causes is necessary. The result of these analysis steps will be the identification of potential tasks to avoid any critical
failures or damage. The result of the analysis can also be that no scheduled maintenance task is required. However, a complete SMA of a product concerning safety relevant failures or damage should be considered mandatory.

The last activity of a SMA analyst is to analyze the identified potential scheduled maintenance tasks to decide which are the most effective and to determine the maintenance intervals or thresholds. Finally this result will be documented in the LSA database as LSA tasks that are linked to the corresponding event threshold or interval.

6 **Thresholds and intervals from scheduled maintenance analysis**

Along with the introduction of an LSA task to perform the maintenance task analysis for the scheduled maintenance activity resulting from SMA, the justifying event will be defined. For scheduled maintenance, an event will be defined that represents the specific threshold. It should be pointed out that the thresholds and the intervals can be of different types.

- Perform Once (PO) thresholds
- Periodical (PE) thresholds/intervals
- Triggers

6.1 **PO threshold**

A maintenance task that is only performed once is characterized by a PO threshold. Typical examples for such tasks can be a check of wheel nuts after a certain distance or a torque of special screws after a certain time of usage.

![PO threshold](image)

**Fig 2 PO threshold**

6.2 **PE threshold or interval**

A maintenance task that is performed repeatedly during the life time of a product is characterized by a PE threshold/interval. Typical examples for these types of tasks can be a periodic inspection or a periodic servicing task, e.g. engine oil change.

![PE threshold](image)

**Fig 3 PE interval**

6.3 **Mixture of PO and PE thresholds**

The common practice of combining PO activities with PE activities can often be found and should be documented carefully in the LSA database. These different activities can be dependent upon each other, because the second maintenance task is driven by the PE interval and will not be performed before the related PO activity.
6.4 Triggers
A more common approach to the fact that intervals or thresholds can be variable over the complete life cycle is the object of a trigger. In general, a trigger is an event which terminates a period of specific scheduled maintenance and starts another one with different intervals or other maintenance activities.

A typical purpose for triggers is the consideration of familiarization, learning phases and phases after the achievement of a certain age of a product, when deterioration effects can force the implementation of a modified scheduled maintenance concept.

7 Thresholds and intervals from other sources
In addition to the identification of scheduled maintenance activities via SMA, there are further sources from which scheduled maintenance can be derived. It should be pointed out the SMA is a rather extensive analysis activity. For that reason, the number of SMA candidates that can be examined in detail is a matter of budget. In addition to SMA results, scheduled maintenance requirements may be identified by:

- Recommendation of the original manufacturers
- Best engineering judgment
- Experience from other projects
- Determination by customer's regulations
- Certification maintenance requirements
- Fatigue related inspections
- Safety engineering requirements
- Recommended scheduled activities during storage time of components
8 LSA tasks

The scheduled or preventive maintenance actions, including the threshold or interval information, will be documented for affected LSA candidates within the LSA database. The tasks will be analyzed in detail for requirements concerning personnel, support equipment, spares, consumables and facilities. The interconnection between the final LSA tasks and the underlying SMA should be established as effectively as possible to guarantee the consistency of data (e.g. IT-based linking of data within the LSA database).

9 Scheduled maintenance harmonization and task packaging process

As the final step, the identified effective and applicable scheduled maintenance actions which are documented as LSA tasks will be analyzed again to identify equal or similar intervals and for grouping. For example, if there is a complicated removal task to gain access to different items, to be inspected, this can influence the final harmonization of preventive and scheduled maintenance. In this case, the goal could be to avoid the repeated performance of this extensive removal activity at different intervals.

In practice, it is necessary to harmonize all the identified scheduled maintenance tasks around a Maintenance, repair and overhaul plan which allows specific values for scheduled maintenance intervals. To find a proper harmonization of scheduled maintenance is challenging for the maintainability engineers because it must be clear that intervals resulting from SMA do not always perfectly fit into the planned intervals in a Maintenance, repair and overhaul plan of the individual customers. For that reason, the adaptation of intervals or thresholds is often required.

It is important to remember that intervals relating to safety relevant maintenance activities cannot be modified without further analysis. The extension of intervals is a very sensitive issue and must not be done without reconsideration of the SMA. Normally, an extension of intervals must correspond to a design change or an alteration of the scheduled maintenance activities. Extensions of intervals for scheduled maintenance concerning economical aspects are also to be investigated carefully. In this case, the risk of a possible economic loss must be balanced. Even more sensitive is the estimated ecological consequences of the extension of intervals for scheduled maintenance and a resulting occurrence of an ecological critical failure. In general you should be aware that the extension of scheduled maintenance intervals is a critical subject.
Also, the shortening of scheduled maintenance intervals cannot be done without deeper investigation. In this case, tasks will be performed more frequently than is really necessary. This will cause additional costs and must be discussed and harmonized with the customer.

As shown in Fig 7, a maintenance task analysis for the complete package should also be performed. Additionally, it must be taken into consideration that the scheduled maintenance packages can contain a large number of single tasks. These tasks will not normally be performed sequentially, but in parallel. All requirements, for example, personnel and support equipment, must be harmonized for the complete package in comparison to the collection of the single LSA tasks.

If a scheduled maintenance package is finalized, each of original tasks, which are written against the corresponding hardware breakdown elements, are no longer of special interest concerning the calculation of logistic values. The original tasks should be maintained. It must be possible, at any stage of a project, to trace the scheduled maintenance package back to its origin tasks from the corresponding SMA and to trace each scheduled maintenance task back to the corresponding SMA (S4000M, MSG-3, and RCM). By following this methodology, you can ensure the complete consistency of the scheduled maintenance from the first analysis steps within the extensive special analysis procedures to LSA tasks and to scheduled maintenance packages within a Maintenance, repair and overhaul plan.
Chapter 11

Level of repair analysis

Table of contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level of repair analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>1</td>
</tr>
<tr>
<td>1 General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>2</td>
</tr>
<tr>
<td>2 LORA Candidate selection aspects</td>
<td>2</td>
</tr>
<tr>
<td>3 Repair or discard decision</td>
<td>3</td>
</tr>
<tr>
<td>4 Identification of optimum maintenance level by ELORA</td>
<td>3</td>
</tr>
<tr>
<td>5 Data gathering for LORA</td>
<td>4</td>
</tr>
<tr>
<td>5.1 Determine parts hierarchy</td>
<td>4</td>
</tr>
<tr>
<td>5.2 Determine LORA candidates</td>
<td>4</td>
</tr>
<tr>
<td>5.3 Collect unit prices</td>
<td>4</td>
</tr>
<tr>
<td>5.4 Collect maintenance related cost data</td>
<td>5</td>
</tr>
<tr>
<td>5.5 Other maintenance relevant data</td>
<td>6</td>
</tr>
<tr>
<td>6 Performance of ELORA and preparation of results</td>
<td>7</td>
</tr>
<tr>
<td>6.1 ELORA baseline run</td>
<td>7</td>
</tr>
<tr>
<td>6.2 Sensitivity analysis</td>
<td>7</td>
</tr>
<tr>
<td>6.3 Maintenance solution recommendation</td>
<td>8</td>
</tr>
<tr>
<td>6.4 LORA Report</td>
<td>8</td>
</tr>
<tr>
<td>6.5 Maintenance solution documentation in the LSA database</td>
<td>8</td>
</tr>
<tr>
<td>7 Example for maintenance level definitions</td>
<td>9</td>
</tr>
<tr>
<td>7.1 General</td>
<td>9</td>
</tr>
<tr>
<td>7.2 Level 1</td>
<td>9</td>
</tr>
<tr>
<td>7.3 Level 2</td>
<td>9</td>
</tr>
<tr>
<td>7.4 Level 3</td>
<td>10</td>
</tr>
</tbody>
</table>

List of tables

<table>
<thead>
<tr>
<th>No.</th>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>References</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Overview of typical cost elements</td>
<td>5</td>
</tr>
</tbody>
</table>

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 3</td>
<td>LSA Business process</td>
</tr>
</tbody>
</table>

Applicable to: All

S3000L-A-11-00-0000-00A-040A-A

Chap 11

DMC-S3000L-A-11-00-0000-00A-040A-A_001_00_EN-US.doc

2010-04-01 Page 1
1 General

1.1 Introduction

Level of Repair Analysis (LORA) is an analysis performed to assist in the establishment of an optimized maintenance solution based on a support philosophy harmonized with the customer. This includes the determination of where repairable LSA candidates are removed, replaced, repaired or discarded. Repair decisions should consider both economic and non-economic factors such as costs, reliability values, maintainability/testability constraints or availability goals for the system. The results of the analysis influence maintenance tasks and corresponding task requirements (eg support equipment, personnel and spare parts).

The two main aspects of the LORA process are the technical assessment and commercial (costs) balancing. The mainly commercial approach, Economical LORA (ELORA), can be performed in conjunction with LCC analysis activities (costs for maintenance activities are a crucial part of the complete life cycle costs). It should be emphasized that only the combination of technical and commercial aspects will lead to a proper LORA result.

The LORA requires a two stage approach:

- Determination of whether the Item under Analysis (IuA) should be repaired or discarded (supporting the repair/discard decision by the customer). This repair/discard decision can include non-economic as well as economic factors.
- Identification of the optimum maintenance level for the required maintenance tasks, also called ELORA.

These two stages are described in more detail in Para 3 and Para 4.

The requirement for a LORA is usually documented in the LSA program plan or similar documents. Depending upon the individual phase of the project, a LORA should be performed for different purposes using different methods.

The LORA process should be repeated as often as necessary during the design and the in-service phases as the design matures to determine whether the maintenance solution remains as predicted or has changed due to the change in design or other factors. The results of the LORA process can also be used to drive supplier selection by illustrating the various options, and could lead to the adoption of phased support.

1.2 Objective

The purpose of a LORA is to provide the customer with decision guidance for the selection of the best balanced maintenance solution. It should be pointed out that different customers may come to individual decisions depending on their intentions or preconditions. In the early stage of a project, LORA aspects must be considered in order to influence the design (eg testability features, modularity, accessibility requirements) as well as to support customer strategy decisions (eg 2-level maintenance strategy, single source principle).

1.3 Scope

This chapter is directed at contractor and customer logistics personnel who perform LORA activities. Results of the LORA will be documented in appropriate LORA reports. The LSA database should be used to document the derived maintenance solution per LSA candidate.

2 LORA Candidate selection aspects

The identification of potential LORA candidates is usually based on the LSA Candidate Item List (CIL), which is established in accordance with the project’s LSA candidate selection process. For details, refer to Chap 3.

As already mentioned above, the LORA is a two step process. The first step in the LORA is to decide whether a LSA candidate is potentially repairable or not. When the item is potentially repairable, a decision must be made on which of these LORA methods is feasible to use:
1. Best engineering judgment
2. Simplified LORA (based on information collection and consideration of values)
3. LORA based on mathematical models (supported by commercial software packages)
4. Simulation (supported by commercial software packages)

In either case, a justification based on best engineering judgment must be performed.

In addition to these factors, other additional constraints or fixed customer decisions that should be considered include:

- Exclusions concerning repair due to danger of destruction or hazardous/unsafe situation
- LORA relevant input by system integrator or Original Equipment Manufacturer (OEM)
- Cut off values or design driven influences (a set of predefined cut off values may help in the reduction of the number of ELORA candidates. These cut off values are usually based on technical and economic factors such as low task frequency, low unit price, repair costs)
- Customer imposed maintenance constraints external to the LORA
- Existing maintenance infrastructure of the customer derived from Operational Requirements Document /Customer Requirements Document (ORD/CRD)
- Customer requirements derived from ORD/CRD
- Contractual requirements or LSA Guidance Conference (LSA GC) decisions
- Clear understanding of the requirements of LORA by other disciplines

These external influences or requirements should be addressed at the LSA GC. Such factors must be agreed upon during the LSA GC, failure to do so may have negative impacts later in the Project.

3 Repair or discard decision

A Repair/Discard decision is the prerequisite to any ELORA. It is, in effect, a screening process in order to minimize the number of ELORA candidates using a logical process of elimination. Each LSA CIL has numerous items that are potential ELORA candidates. The Repair/Discard decision must be carried out in order to identify which of these items are candidates for further ELORA analysis. This is achieved by determination of the repair feasibility for the items.

For each potential item, crucial technical questions must be clarified first:

- Does the design of the item allow repairs in general? If so, this item is a potential ELORA candidate.
- Is the repair of the item limited to certain maintenance levels? If so, this item is a potential ELORA candidate. The analysis must be limited to the possible maintenance levels.
- Does the design of the item disallow repairs in general? If so, this item must be removed from the ELORA candidate list. However, the proper maintenance level for replace and discard must be identified and documented carefully.

This decision, and the logic leading to this decision, must be recorded in the LSA database or similar database for future traceability. For items that have been identified as potentially repairable, the next stage is to further determine the method of performing the ELORA. This stage presents the opportunity to have any external influences/requirements taken into account (eg existing maintenance solutions used by the customer, cut off values or design driven influences).

4 Identification of optimum maintenance level by ELORA

The repair/discard decision process will result in a list of remaining ELORA candidates. These candidates are identified as potentially repairable. The objective of the ELORA is to determine the optimum maintenance solution for these candidates. This is carried out using mainly economic cost factors such as spare parts, support equipment required to diagnose and perform repairs, personnel, training, facilities, technical documentation, Packaging, Handling,
Storage and Transportation (PHST), and also non-economic factors such as availability requirements, reliability, and maintainability can be taken into account.

There are many different mathematical ELORA models available and supported by different commercial software packages. Alternatively, a project specific or tailored model can be used. A project specific or tailored model might allow better consideration of customer requirements, constraints (both customer and OEM), available infrastructure, etc. This approach is often used within a simplified LORA.

It is of vital importance that all LORA analyses within a project use the same LORA model. Failure to do so will lead to differing results.

5 Data gathering for LORA
The required data for a LORA should be collected carefully. A LORA data set can contain simple data such as whether or not there is a need for:

- Spare parts
- Facilities
- Support equipment required to diagnose and perform repairs
- Personnel and required skill levels
- Extended training

But a LORA data set can also contain comprehensive data giving in-depth details of the requirements and especially the costs associated with those requirements.

Once the data elements have been agreed upon in the LSA GC, the LORA data can be gathered from a variety of sources such as drawings, technical documents, contracts, supplier information, commercial department, etc. The following paragraphs detail some of the data gathering tasks.

5.1 Determine parts hierarchy
Failure modes in LORA are assigned to the hierarchical structure of the system. Failures at the top level of the system are assumed to be primarily caused by failures of the items (or subsystems) at the next lower indenture level. Similarly, failures of the items in the 2nd level of indenture are mainly initiated by failures of the items in the next lower indenture level. For this reason, it is impractical to perform a LORA until hierarchical relationships are defined. For systems that are currently in production, the hierarchy and subsequent LORA candidate items should be easily determined from existing LSA data concerning system breakdown.

5.2 Determine LORA candidates
After the parts hierarchy is established, the analyst must determine which parts will be considered in the analysis. On the surface it might seem logical to include all of the parts, however, inclusion of non repairable or consumable parts such as nuts, bolts, gaskets, etc, will add no value to the analysis. Candidate items can be determined via a rule set based on existing repair or maintenance codes, unit price, repair level, or other criteria.

5.3 Collect unit prices
Unit prices for all items are required to fill an ELORA data model and to perform the analysis. Sources for unit prices can be LSA databases, provisioning records or other sources. Unit price for all ELORA candidate items is necessary to determine the economic feasibility of repair. It must be ensured that for ELORA calculation purposes, the correct values for unit prices will be used. It should be noted that unit prices can differ dramatically from the production phase to the in-service phase (spare part price can be very different from production unit price). Sometimes precise values are difficult to obtain and are therefore estimated.
5.4 Collect maintenance related cost data

Maintenance cost data includes labor time, training, spare parts and consumables, support equipment, facilities, technical documentation and PHST aspects. Concerning these data it must be distinguished between initial and recurring costs. These data have to be used in the calculation of cost at each possible maintenance level.

The following table gives an overview of which typical types of cost elements should be considered in an ELORA (additional data can be required depending on the Project).

<table>
<thead>
<tr>
<th>Table 2  Overview of typical cost elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cost element</strong></td>
</tr>
<tr>
<td><strong>Personnel</strong></td>
</tr>
<tr>
<td>Labor time costs</td>
</tr>
<tr>
<td>Training costs (teachers and staff)</td>
</tr>
<tr>
<td>Training equipment costs (eg simulators, classroom equipment, training facilities)</td>
</tr>
<tr>
<td>It should be pointed out that not only direct labor should be considered, but also required indirect labor (eg line managers, inspectors, work preparation department)</td>
</tr>
<tr>
<td><strong>Spare parts / consumables</strong></td>
</tr>
<tr>
<td>Spare parts / consumables provision</td>
</tr>
<tr>
<td>Spare parts / consumables storage</td>
</tr>
<tr>
<td>Disposal costs</td>
</tr>
<tr>
<td><strong>Support equipment to diagnose and repair</strong></td>
</tr>
<tr>
<td>Support equipment provision and replacement</td>
</tr>
<tr>
<td>Support equipment maintenance</td>
</tr>
<tr>
<td>Support equipment upgrade</td>
</tr>
<tr>
<td>Development of customized software (eg test software)</td>
</tr>
<tr>
<td>Support of customized software for support equipment</td>
</tr>
<tr>
<td>Disposal costs</td>
</tr>
<tr>
<td><strong>Facilities and infrastructure</strong></td>
</tr>
<tr>
<td>Building of facilities and infrastructure</td>
</tr>
<tr>
<td>Maintenance of facilities and infrastructure</td>
</tr>
<tr>
<td>Costs of operation for facilities and infrastructure</td>
</tr>
<tr>
<td>Costs for modification of facilities and infrastructure</td>
</tr>
<tr>
<td><strong>Technical documentation</strong></td>
</tr>
<tr>
<td>Documentation costs (eg user handbooks, maintenance and training manuals, illustrations, spare part catalogues)</td>
</tr>
<tr>
<td><strong>PHST aspects</strong></td>
</tr>
<tr>
<td>Packaging costs</td>
</tr>
<tr>
<td>Stock keeping costs (eg maintenance tasks and handling during storage, special storage containers, administration of stock)</td>
</tr>
<tr>
<td>Transportation costs (transport to maintenance facilities and back to user)</td>
</tr>
<tr>
<td>Cost element</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Others</td>
</tr>
<tr>
<td>Average repair costs at industry</td>
</tr>
<tr>
<td>Additional costs (documentation, administration, preparatory work and post processing)</td>
</tr>
<tr>
<td>Cost per reorder action</td>
</tr>
<tr>
<td>Cost per requisition</td>
</tr>
<tr>
<td>Disposal costs</td>
</tr>
<tr>
<td>Discount rate</td>
</tr>
<tr>
<td>Holding cost percentage</td>
</tr>
<tr>
<td>Interest rate</td>
</tr>
</tbody>
</table>

### 5.5 Other maintenance relevant data

Besides the direct costs related data there is other information of interest for ELORA calculations. The following list gives an overview of data that should be considered in an ELORA.

#### Time and duration information

- Reliability information (Mean Time Between Failure (MTBF), Mean Time between Unscheduled Removal)
- Task frequency of corrective maintenance task
- Scheduled maintenance interval
- Task duration
- Logistics down time (waiting time for logistic resources)

**Note**

Data for repair time is often difficult to obtain, especially for new systems. LSA databases sometimes contain this information. Some developmental systems specify desired and maximum times to repair a failure. This can be used as a repair time for items that are directly removed and replaced. For the repair of lower indenture items, it will probably be necessary to use the repair time for a similar item on another system.

#### Availability values

- Minimum availability of IuA (usually, operational availability is specified)
- Availability of spare parts in stock
- Availability rate of personnel for maintenance activities

#### Stock relevant values

- Procurement lead time
- Pipeline transit times
- Repair turnaround times

#### General information

- Number and type of contractor industrial facilities
- Number of operational sites
- Number of system operated per site
- Distances between sites
- Inflation/discount rate
- Repair policy (e.g., 2-level maintenance strategy)
History shows that the majority of cost influencing decisions must be made in a very early stage of the project, even if the information is very limited and the design is not yet stable. This emphasizes that the LORA must be repeated throughout the life cycle so that the requirement for adoption of the maintenance solution can be identified in a timely manner.

6 Performance of ELORA and preparation of results

After successful collection and harmonization of data with the Customer, the ELORA is performed. This can be done simply by balancing the given information (best engineering judgment). However, when using complex mathematical calculation models, a commercial software package may be selected. The data set to be collected depends upon the requirements of the specific software package. The results of the calculation runs should be prepared in an ELORA report and distributed according to established rules.

6.1 ELORA baseline run

The first step of an ELORA, using a supporting software package, is the baseline run based on the gathered ELORA data. This baseline run will result in a first set of information regarding the cost situation for the individual maintenance levels. The result can be clear cut or an ambivalent situation between the different maintenance levels. In either case, it is recommended to confirm the baseline run results by executing a sensitivity analysis.

6.2 Sensitivity analysis

A sensitivity analysis should be performed on various cost significant parameters. The results of the various sensitivity runs will indicate whether the maintenance solution from the baseline run is stable or not. Parameters used in a sensitivity analyses include:

- Major cost influencing parameters of the IuA itself (eg unit price, MTBF)
- Any parameters that are critical to cost intensive logistic requirements (eg facilities, infrastructure, support equipment, highly educated personnel)
- Any parameters using assumptions or estimates because there was no data available (eg historic or similar data from other products)

Once the parameters for the sensitivity analyses have been selected, the next step is to establish an appropriate numerical range for the parameter. Finally, when all the requirements have been established, an ELORA is performed for each parameter variation. Changing single parameters will keep the number of runs to a minimum. Multiple parameter changes at the same time will lead to nested runs and therefore should be avoided because evaluation of the results can be confusing.

The output of the sensitivity analyses can be used as a baseline to establish a preliminary maintenance solution. Additionally, it can be used to influence other logistic disciplines and/or design, in order to change their input to drive the equipment to enable an intended maintenance solution.

A sensitivity analysis is usually conducted to complement baseline results. Sensitivity runs are performed for cost significant parameters that have a low confidence level or to accomplish a trade-off study. For a sensitivity analysis of low confidence values, the most common method is to perform runs using the worst and best case data. If there is no shift in maintenance solution result between these two runs, then no additional executions using intermediate values are required.

An example of this is sensitivity for the MTBF of an item. Suppose an estimated value of 50000 hours is used in the baseline run. Preliminary analysis revealed that MTBF’s of similar items range from 15000 hours to 150000 hours. The first sensitivity runs performed would include values of 15000 and 150000 hours for the MTBF of this item. If the most cost effective repair level for both sensitivity runs is the same as the baseline run, no further sensitivity analysis for this MTBF is necessary. However, if a change in repair level occurs for one or both runs,
additional executions using values of MTBF between the high and low value may be necessary to identify the break points.

6.3 **Maintenance solution recommendation**

For a concluding maintenance solution recommendation, it is necessary to consider the cost aspects resulting from an ELORA, as well as technical aspects that can have a major impact on the recommendation of the maintenance solution to the customer (e.g., critical repairs that might destroy the IuA, dangerous repair tasks). Additionally, the customer's preferences should be considered.

It is also important to harmonize the maintenance activities in order to avoid a split of similar maintenance activities to different maintenance sites. For example, the allocation of a piece of electronic equipment for replacement and subsequent repair of different components should not be split between industry and customer industrial sites.

6.4 **LORA Report**

A report documenting the analysis results must be generated. The LORA report should contain, as a minimum, the following:

- Agreements reached between the customer and/or the contractors regarding LORA
- A brief description of the system/equipment under analysis
- Comprehensive list of any assumptions and estimations including their rationale
- Data sources
- Comprehensive list of the LORA data elements
- Input values
  - Operational environment data used for the process
  - Breakdown of IuA
  - Cost and maintenance data
- Realization of mathematical methods applied within a simplified ELORA
- Utilization of ELORA software package
- Result of baseline run
- Results and explanations of sensitivity analysis runs
- Trade-off studies
- Consolidation of technical and cost aspects
- Concluding recommendation of the maintenance solution including discard decision

It is recommended to develop the report structure and layout early in the analysis process. Important information and logic can be overlooked during the course of a lengthy analysis.

For good traceability, it is recommended to use a standardized format within the LORA report for the maintenance solution recommendation and related customer decision. The customer decision will be documented in the LSA database.

6.5 **Maintenance solution documentation in the LSA database**

The content of the maintenance solution is related to the corresponding maintenance activities as documented in maintenance tasks in the LSA database. Each task for an LSA candidate should be provided with attributes that reflect the maintenance solution. This information should contain, as a minimum, the following:

- Is the maintenance task to be performed scheduled or unscheduled?
- For scheduled tasks: interval or threshold data.
- Is the maintenance task to be performed "on" or "off" product?
- The maintenance level at which the task must be performed.
- The place at which the maintenance tasks are to be performed.
− Special remarks or warnings (e.g., when the product is used under special circumstances, inspection interval has to be reduced from 6 to 3 months)
− References to standard repair tasks
− Data source information for traceability (e.g., identification of LORA report)

It is recommended to agree to the basic data set that reflects the maintenance concept during the LSA GC, in order to enable a common understanding. Proper documentation within the LSA database enables a strong reporting ability for different aspects of the maintenance concept.

Examples:
− Complete overview of the maintenance activities at all maintenance levels
− Summary of maintenance activities performed at certain maintenance levels
− Overview of scheduled/unscheduled maintenance activities
− Expected effort at certain maintenance levels
− Support for quality checks concerning completeness and conclusiveness of the maintenance solution

7 Example for maintenance level definitions

7.1 General
An example of a maintenance strategy is given in the following subchapters, based on three levels of maintenance, which indicates the capability of personnel, availability of special facilities, time limits and the environmental conditions to be assumed in determining the functions to be accomplished at each maintenance level.

7.2 Level 1
The goal of level 1 maintenance is to allow the product to remain available. This implies a fast and easy exchange of LRU’s and/or the replacement of modules, performed on the product by organizational personnel when a mal-function occurs.

Level 1 activities are expected to include:
− Servicing activities
− Usage preparation and role changes
− Pre- and post- inspections
− Functional checks
− Trouble shooting
− Preventive maintenance
− Corrective maintenance (repair by replacement and system adjustment)
− Loading of software (operational and engineering) and data retrieval
− Simple modifications

7.3 Level 2
The goal of Level 2 maintenance is to maintain the highest possible level of availability. Operating site maintenance activities are extended to the repair of subassemblies, modules and LRUs after their replacement at maintenance Level 1. Testing on test-benches or integration tests could be performed. Subsequently, Level 2 maintenance can be performed either on or off product.

Level 2 activities are expected to include:
− Repairs down to module and subassembly level
− Moderate structural repairs
− Major scheduled inspections
− Moderate modifications
− Technical assistance to the Level 1 organization
− Software servicing concerning engineering data
− Preservation of complete product

Level 2 activities include corrective and preventive maintenance and specific maintenance activities that will be performed both on product (during maintenance the product will be unavailable for use), and off product. Level 2 includes tasks beyond those accomplished at Level 1 in order to facilitate the return of the equipment to its full operational state. Level 2 maintenance will be performed at facilities capable of accommodating the maintenance tasks, which could include special equipment or specialized workshops, and will be performed by appropriately trained and specialized personnel.

7.4 Level 3
The goal of Level 3 maintenance is to assure the highest possible availability of the product, as well as providing engineering support for operational aspects. All repairs and overhaul activities beyond Level 1 and Level 2 capabilities shall be assured. Major modifications to improve the design and/or operational activities will be prepared and, if necessary, executed at this level.

Level 3 activities are expected to include:
− Repairs down to full reconditioning
− Repairs requiring special or rare skills or support equipment
− Major structural repairs
− Major scheduled inspections
− Extended modifications and update programs
− Technical assistance to the Level 1 and Level 2 organizations
− Software modification
− Preservation of the complete product

Level 3 maintenance shall assure a maximum of autonomy for the user organizations. International cooperation may allow the set-up of an effective and economic authorized level of maintenance. Single source repair should be considered the best solution.

Level 3 maintenance will be performed in appropriately equipped facilities or component repair facilities of the Customer or the Contractor (or subcontractors). Level 3 requires appropriately trained and specialized personnel. Level 3 maintenance also involves the return of defective items (suspected or confirmed) to the OEM for repair/overhaul/retest.

Note
The Level 3 activities as the "highest" maintenance level can also be subdivided into two or more levels (eg Level 3 and Level 4). This can be used when required to clearly separate the user operated activities from the contractor operated activities at industry facilities. In this case, the maintenance strategy would contain four levels of maintenance (or possibly more).
Chapter 12

Maintenance task analysis

Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance task analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Terms, abbreviations and acronyms</td>
<td>3</td>
</tr>
<tr>
<td>2 Task justification</td>
<td>4</td>
</tr>
<tr>
<td>3 Categorization of activities</td>
<td>5</td>
</tr>
<tr>
<td>4 Task documentation</td>
<td>6</td>
</tr>
<tr>
<td>4.1 General aspects</td>
<td>6</td>
</tr>
<tr>
<td>4.2 Practical considerations</td>
<td>6</td>
</tr>
<tr>
<td>5 Task structure</td>
<td>9</td>
</tr>
<tr>
<td>5.1 Rectifying and supporting tasks</td>
<td>9</td>
</tr>
<tr>
<td>5.2 Structure of supporting tasks</td>
<td>10</td>
</tr>
<tr>
<td>5.3 Structure of a rectifying task - referencing method</td>
<td>11</td>
</tr>
<tr>
<td>5.4 Task preconditions, pre-work and post-work</td>
<td>12</td>
</tr>
<tr>
<td>5.5 Narrative description</td>
<td>13</td>
</tr>
<tr>
<td>6 Task frequency</td>
<td>13</td>
</tr>
<tr>
<td>6.1 Maintenance activities due to inherent equipment failures</td>
<td>13</td>
</tr>
<tr>
<td>6.2 Task frequency for supporting tasks</td>
<td>14</td>
</tr>
<tr>
<td>6.3 Maintenance activities due to damages or special events</td>
<td>15</td>
</tr>
<tr>
<td>6.4 Scheduled maintenance activities</td>
<td>16</td>
</tr>
<tr>
<td>7 Task resources and task duration</td>
<td>16</td>
</tr>
<tr>
<td>7.1 Resources</td>
<td>16</td>
</tr>
<tr>
<td>7.2 Resources out of references</td>
<td>18</td>
</tr>
<tr>
<td>7.3 Harmonization of support equipment and spare parts</td>
<td>19</td>
</tr>
<tr>
<td>7.4 Task location aspects</td>
<td>19</td>
</tr>
<tr>
<td>7.4.1 Location in conjunction with the product itself</td>
<td>19</td>
</tr>
<tr>
<td>7.4.2 Location in conjunction with the required facility/infrastructure</td>
<td>19</td>
</tr>
<tr>
<td>7.4.3 Location in conjunction with the zone within the product itself</td>
<td>19</td>
</tr>
<tr>
<td>7.5 Product and system availability during maintenance performance</td>
<td>19</td>
</tr>
<tr>
<td>7.6 Support solutions (task variants)</td>
<td>20</td>
</tr>
<tr>
<td>7.7 Task duration</td>
<td>20</td>
</tr>
<tr>
<td>7.8 Parallel activities within maintenance tasks</td>
<td>21</td>
</tr>
<tr>
<td>8 Training requirements</td>
<td>23</td>
</tr>
<tr>
<td>9 Examples of assigning rectifying tasks to LSA candidates</td>
<td>23</td>
</tr>
<tr>
<td>9.1 Example 1: Major failure of equipment, non-repairable</td>
<td>25</td>
</tr>
<tr>
<td>9.2 Example 2: Mainboard failure of equipment 401 - Option 1</td>
<td>27</td>
</tr>
<tr>
<td>9.3 Example 3: Mainboard failure of equipment 401 - Option 2</td>
<td>28</td>
</tr>
<tr>
<td>9.4 Example 4: Failure of device 01, repairable at operational site</td>
<td>30</td>
</tr>
<tr>
<td>9.5 Example 5: Complex maintenance procedure on several levels</td>
<td>32</td>
</tr>
</tbody>
</table>

List of tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>References</td>
</tr>
</tbody>
</table>

List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
</table>

Applicable to: All
List of figures

1. Event-task correlation ................................................................. 4
2. General equipment example ...................................................... 6
3. Remove procedure including preliminary work .......................... 10
4. Remove procedure without preliminary work activities ............. 11
5. Typical rectifying task - repair procedure ................................. 12
6. Task resources .......................................................................... 18
7. Parallel activities and their resources/durations ........................... 22
8. Equipment example ................................................................. 23
9. Breakdown of Equipment 401 ................................................... 24
10. Major failure of equipment, non-repairable .............................. 25
11. Schematical LSA representation of example 1 ........................... 26
12. Mainboard failure of equipment 401 - rectified by equipment 401 replacement ............................................ 27
13. Schematical LSA representation of example 2 ........................... 28
14. Mainboard failure of equipment 401 - rectified by equipment 401 repair ............................................................... 28
15. Schematical LSA representation of example 3 ........................... 29
16. Failure of device 01, repairable at operational site .................... 30
17. Schematical LSA representation of example 4 ............................ 31
18. Major failure of equipment, only repairable at higher level ........ 32
19. Schematical LSA representation of example 5 ............................ 34

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 7</td>
<td>LSA failure modes and effects analysis</td>
</tr>
<tr>
<td>Chap 8</td>
<td>Damage and special event analysis</td>
</tr>
<tr>
<td>Chap 9</td>
<td>Logistic related operations analysis</td>
</tr>
</tbody>
</table>

Applicable to: All  
S3000L-A-12-00-0000-00A-040A-A  
Chap 12

DMC-S3000L-A-12-00-0000-00A-040A-A_001_00_EN-US.doc  
2010-04-01 Page 2
1 General

1.1 Introduction

The requirement for a maintenance task has to be analyzed from different perspectives. The first step will be to divide up the entire task into appropriate working steps, which will be called subtasks. This is necessary to apply a useful structure to a complex and time-consuming activity. It is not unusual for repair or replace procedures to contain hundreds of subtasks. The sequence of each of the subtasks must be clear. The decision to divide up an activity depends on the requirements for the depth of information. Depending on the use case, it may be sufficient to break down an activity into 10 subtasks with some general information. In other use cases, where there is a requirement to perform a deep analysis, the same task, which was broken down in just 10 subtasks before is now divided up into 40 subtasks. Additionally, an analysis of the tasks concerning the resources is required. Everything that is needed to perform the maintenance task must be identified within the Maintenance Task Analysis (MTA).

1.2 Objective

This chapter is a guideline for the analysis of an identified maintenance task concerning its logistic requirements. This mainly includes spare parts and consumables, support equipment, personnel, facilities and task duration information. Additional information such as task criticality, task location, training needs, pre- and post-conditions or safety requirements should also be taken into consideration.

1.3 Scope

This chapter is addressed to analysts who perform analysis activities within the LSA process to identify corrective and preventive maintenance tasks. After the identification, a deeper analysis of the maintenance activities is required. It is recommended that the establishment of close co-operation between disciplines such as maintainability, testability and reliability, together with the personnel to perform an MTA, is created. An ideal solution would be to combine these duties and assign them to one analyst.

1.4 Terms, abbreviations and acronyms

The following specific terms, abbreviations and acronyms used within this chapter. More special terms are explained directly within the appropriate paragraphs. Refer to Chap 21 for a full listing of Terms, abbreviations and acronyms.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supporting task</td>
<td>A supporting task is a part of a complete maintenance activity. As a standalone task it cannot rectify an event, such as failures, damages, special events or thresholds. A supporting task contains subtasks in terms of working steps.</td>
</tr>
<tr>
<td>Rectifying task</td>
<td>A rectifying task is any maintenance activity that resolves an event, such as failures, damages, special events or thresholds. A rectifying task contains subtasks in terms of referenced supporting tasks and/or definite working steps.</td>
</tr>
</tbody>
</table>
2 Task justification

Any maintenance task or operational activity that is identified as required will cause effort within support engineering and logistic disciplines. For that reason there should be a rational justification for the performance of each activity in the environment of maintenance and operational tasks. The principle of event driven maintenance is one of the main aspects.

Maintenance tasks and related operational tasks to be considered by an LSA are justified by corresponding logistic analysis activities. Failures, damages, special events and thresholds are the primary drivers for maintenance activities and are described in their specific chapters.

Refer to:
- Chap 7 LSA FMEA
- Chap 8 Damage and special event analysis
- Chap 10 Schedule Maintenance Analysis

Additionally, there are additional general tasks which originate from other sources (eg handling and servicing requirements or software and data loading).

Refer to:
- Chap 9 Logistics Related Operations Analysis

For proper documentation of the connection between the justifying source and the related maintenance activity a corresponding structure in the LSA database should be established. The following figure gives an overview of the relationship between maintenance/operational tasks and their justification.

![Event-task correlation diagram](ICN-B6865-S3000L0049-001-01)

**Fig 1 Event-task correlation**

Especially in the area of operation tasks or data/software loading, it can be difficult to address exactly whether the task is a maintenance task with a clear justification by a maintenance relevant event or the task is a general supporting task.

Example: The towing of an aircraft can have different reasons:
- Towing out of the hangar to prepare for a mission
- Towing for refueling purposes
- Towing to the maintenance hangar in case of a failure
In this case, it is not possible to indicate clearly whether the towing task is part of a maintenance activity or an operational activity. However, this task must be considered as a specific supporting task, which is required for different needs. Documentation of these activities and the question of who is responsible for them must be agreed upon between the customer and the contractor, within the LSA guidance conference, in order to avoid later misunderstanding or even the neglect of such activities. The integration of these activities within the system breakdown is described in Chap 3.

3 Categorization of activities

For unique task identification, the use a coding system such as that given in S1000D for technical publication is recommended. The activity is clearly categorized in two data elements, information code and information code definition. Taking into account that technical documentation and the LSA are strongly interconnected, it is recommended that the use of the same task categorization method in the LSA database is considered. This avoids the requirement for a translation matrix from the LSA task coding and nomenclature to the coding and naming within a technical documentation system. This approach improves communality, traceability and, in the end, ILS quality.

In the description of a code for a special activity, there must always be an appropriate verb, which clearly identifies the activity. Avoid the usage of general terms. In the following table you will find some examples from S1000D.

<table>
<thead>
<tr>
<th>Information code</th>
<th>Information code description</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>Clean</td>
</tr>
<tr>
<td>251</td>
<td>Clean with chemical agents</td>
</tr>
<tr>
<td>256</td>
<td>Polish and apply wax</td>
</tr>
<tr>
<td>341</td>
<td>Manual test</td>
</tr>
<tr>
<td>342</td>
<td>Automatic test</td>
</tr>
<tr>
<td>520</td>
<td>Remove procedure</td>
</tr>
<tr>
<td>720</td>
<td>Install procedure</td>
</tr>
<tr>
<td>752</td>
<td>Data loading</td>
</tr>
<tr>
<td>921</td>
<td>Remove and replace procedure</td>
</tr>
</tbody>
</table>

Note

With the help of the information code of S1000D, activity groups are defined. There are more general information codes, such as 250 for clean. Additionally, more specific procedures concerning cleaning activities can be applied with a more detailed information code such as 251 for cleaning with chemical agents.
Task documentation

4.1 General aspects
The selection criterion for LSA candidate items is described in Chap. 3. Maintenance concepts for breakdown elements selected as LSA candidates can be documented in different ways. This paragraph should sensitize the analyst to proper selection of LSA candidates and the corresponding task documentation method. For these purposes, a very simple general example is given in Fig 2 to begin clarification of the different methods that can theoretically be used.

![Diagram of an assembly with parts 01 to n](image)

Fig 2 General equipment example

The assembly in Fig 2 above contains n different parts. Theoretically, the assembly itself and each part within the assembly can fulfill criteria to be selected as a potential LSA candidate. From a maintenance point of view, there are different possibilities by which to document the identification of maintenance tasks concerning the repair of the assembly:

<table>
<thead>
<tr>
<th>Assembly view</th>
<th>Breakdown view</th>
</tr>
</thead>
<tbody>
<tr>
<td>One LSA candidate selected = assembly itself, eg equipment</td>
<td>n LSA candidates selected (each part could become an LSA candidate, because it can be replaced)</td>
</tr>
<tr>
<td>n maintenance tasks to be documented: repair assembly by replacement of part 1 to part n</td>
<td>n maintenance tasks to be documented: replace part 1 to part n</td>
</tr>
</tbody>
</table>

Table 4 shows the extreme situation of an either-or decision. As the documentation of an LSA candidate should be considered as a cost intensive activity, it is recommended to be economical with addressing breakdown elements to be an LSA candidate.

4.2 Practical considerations
The situation in reality is more complex than described in Para 4.1. The parts from the simple example above can be repairable themselves or be discard items. Being repairable can make a
specific part within a piece of equipment an LSA candidate also. In this situation, having a LSA candidate Item within another LSA candidate Item cannot always be avoided.

One of the decisive questions will be which component of the equipment has what kind of properties concerning the possibility of repair at which maintenance level. This has a large impact on the storekeeping method of the different items. To clarify terminology, it should be distinguished between several types of repairable.

### Table 5 Component types

<table>
<thead>
<tr>
<th>Component type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-repairable</td>
<td>A part that will not be repaired at any maintenance level. The reason can be technical and/or economical. The methods of replacement must be clarified.</td>
</tr>
<tr>
<td>Repairable on industry maintenance level</td>
<td>An equipment/part which can be repaired only on industry level. For such items, only the replacement will be described in the LSA database. It is recommended to identify the possibility of repair on industry level in the LSA database; a detailed description of repair tasks is not required. The ways of replacement within the storekeeping must be clarified.</td>
</tr>
<tr>
<td>Repairable on customer maintenance level, depot site</td>
<td>A part/equipment that can be repaired on a specialized customer level. For such items, the replacement tasks and the repair procedures can be described in the LSA database. The level of technical documentation for the maintenance activities should be clearly defined within the LSA guidance conference. The methods of replacement within the storekeeping must be clarified.</td>
</tr>
<tr>
<td>Repairable at customer maintenance level, operational site</td>
<td>A part/equipment that can be repaired at customer level at an operational site. For such items, the replacement tasks and the repair procedures should be described in the LSA database. The level of technical documentation for the maintenance activities should be defined clearly within the LSA guidance conference. The methods of replacement within the storekeeping must be clarified.</td>
</tr>
</tbody>
</table>

This rough classification of equipment/parts additionally depends on the influencing events on the items. There can be failures that lead to the disposal of the complete equipment/part, and other failures will result in the equipment/part being repaired. Each event has to be analyzed to identify which maintenance activity at which maintenance level must follow to rectify the event. It should be pointed out that a definitive decision is not always possible. There will be events that can be rectified on several maintenance levels. In this case, not only the types of influencing events are significant, but also the economic aspects must be considered carefully. For example the result of a Level of Repair Analysis (LORA) can be the replacement of the complete equipment (with a following discard) because of economic aspects, however the equipment would be repairable. Also, mixed scenarios are possible. For example, a certain percentage of a specific failure is rectified at the customer maintenance level, and the balance will be returned to the industry maintenance level because of capacity constraints at customer maintenance level.

In practice, many different situations concerning maintenance driving events (especially failures) and the following required chain of maintenance activities can occur. For better understanding,
Table 6 gives an overview of typical examples. Additionally, an example of a rather simple piece of equipment is given in Para 9 (different maintenance situations and how they should be documented in the LSA).

### Table 6  Typical situations of task documentation requirements

<table>
<thead>
<tr>
<th>Event</th>
<th>Rectifying task</th>
<th>Possible following task</th>
<th>Remark</th>
</tr>
</thead>
</table>
| Failure of a non-repairable component of equipment | Repair equipment by replacement of faulty component at operational site | Disposal of faulty component | Non-repairable component is required as a spare part  
If the documentation of the discard of the faulty component is required, a specific disposal procedure would be documented against the component (as the appropriate LSA candidate) as a separated rectifying task |
| Failure of a repairable component of an equipment | Repair equipment by replacement of faulty component at operational site | Repair faulty component at operational site | End item is not waiting until the completion of the repair of the component  
Component is required as a spare part |
| Failure of a repairable component of an equipment | Repair equipment by direct repair of faulty component at operational site | None | End item is waiting until the completion of the repair of the component  
Component is not required as a spare part, because the repaired component will be re-installed |
| Failure of repairable component of an equipment, component is repairable at customer depot site | Repair equipment by replacement of faulty component at operational site | Forwarding component to customer depot site  
Repair faulty component at customer depot site | Repairable component is required as a spare part at operational site  
If the documentation of the repair of the faulty component as a following task is required, it would be documented against the component (as the appropriate LSA candidate) for the repair |
| Failure of repairable component of an equipment, component is repairable at industry level | Repair equipment by replacement of faulty component at operational site | Forwarding component to industry | Repairable component is required as a spare part at operational site  
Documentation of repair on industry level not required |
| Failure of equipment, only repairable at industry or customer depot site | Replace faulty equipment at operational site | Forwarding equipment to industry  
Repair equipment at industry or customer depot site | Equipment is required as a spare part at operational site  
If the documentation of the repair of the faulty equipment at customer depot site as a following task is required, it would be documented against the equipment (as the appropriate LSA candidate) for the repair |
It is not possible to document all combinations of events and the following repair/replace activities within this chapter. The examples given are typical ones. Depending on additional aspects (e.g., abilities at the different maintenance levels, scrap rates of equipments or components, end item waiting for repair or not), the sequence of activities can be influenced in multiple ways. The examples should help clarify the principle of summarizing maintenance activities within a few full LSA candidates. A number of following tasks on higher maintenance levels need not to be analyzed in detail. Only the identification of such possible tasks can be important. In most cases, a failure of equipment is rectified by two different task types:

- Replacement of the entire equipment (represented by a specific breakdown element)
- Repair of equipment by replacement of parts (as a following task)

The following tasks describe the repair procedures that must be carefully analyzed in order to decide where these activities must be linked to and how deep of a detailed description is required. Another aspect to be considered is that not every replacement task can be performed at a customer operational site. For this reason, it may be necessary to transport the complete product/end item itself to a higher level maintenance site. In this case, the replacement of defective equipment will be performed at industry level or at a customer depot site.

5 Task structure

Previous experience has shown that a common understanding of how to create a proper maintenance task structure may not be accomplished without any problems. For this reason, the following paragraphs give an introduction of how to categorize maintenance activities, how to introduce a hierarchy of tasks and subtasks and how to use information within existing tasks efficiently with the help of an intelligent referencing system.

5.1 Rectifying and supporting tasks

One important step to give a structure to the tasks is to divide them into two classes of tasks.

- Rectifying tasks ("event solver")
- Supporting tasks

Each maintenance activity is driven by an event. This event can be a failure, damage, a special event or a deadline, which has expired. All of these events require a maintenance action to be completed. Each task that is able to resolve such an event should be defined as a rectifying task. Typical examples for rectifying tasks are the following:

<table>
<thead>
<tr>
<th>Event</th>
<th>Rectifying tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure</td>
<td>Repair or replace procedures</td>
</tr>
<tr>
<td>Damage</td>
<td>Repair or replace procedures</td>
</tr>
</tbody>
</table>
### Event Rectifying tasks

<table>
<thead>
<tr>
<th>Event</th>
<th>Rectifying tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special event</td>
<td>Inspection/fault location</td>
</tr>
<tr>
<td>Threshold</td>
<td>Scheduled maintenance action (eg scheduled replace of a time change item)</td>
</tr>
</tbody>
</table>

As opposed to the rectifying task, a supporting task cannot be the solving task for an event. The supporting task as a standalone task can never resolve a failure or damage and can never be the standalone activity after a special event. Also, operational needs can cause specific activities which belong to the group of supporting tasks. Typical supporting tasks are the following:

- Test procedures
- Gain access/undo access
- Remove and install procedures
- Assemble and disassemble procedures
- Jacking up of a vehicle

It should be clear that even supporting tasks can be complex and time-consuming, therefore, it must be possible to divide these tasks into subtasks.

### 5.2 Structure of supporting tasks

With the help of supporting tasks, the analyst is able to create a library of task modules, which can be used later to support the creation of the more extended rectifying tasks. The supporting task can be divided into work steps. In the following figures, you can see a simple approach for the arrangement of a rather simple remove procedure:

As shown in Fig 3, this example contains both the pre-work and the working steps indicating that the pre-work steps must be completed before the real work concerning the IuA begins. This situation should be avoided when building supporting tasks. All of the preconditions to perform the working steps concerning the IuA itself should be considered as completed when starting with the first working step. For this reason, the situation shown in Fig 3 reduces to a task structure described in Fig 4.
Remove procedure (for equipment 401)

Working step 1: Remove electrical connector E01-013
Working step 2: Open safety screws S01 and S02
Working step 3: Open attaching screws A01 to A07
Working step 4: Remove equipment 401 from housing

Activities, directly connected to the equipment to be removed

ICN-B6865-S3000L0052-001-01

Fig 4 Remove procedure without preliminary work activities

This second way of documenting the supporting tasks is the recommended one. The working steps should be described with the help of subtasks within the supporting task. References to other existing supporting tasks must not be used in order to avoid nested references.

5.3 Structure of a rectifying task - referencing method

After the creation of a library of supporting tasks, the next step will be to arrange the rectifying maintenance tasks, this means those procedures that can serve as "event solvers". In Fig 5, the structure of a typical repair procedure of equipment is shown. Within this repair procedure, there are a number of subtasks. Some of them are subtasks that have already been described on another breakdown indenture level (eg the removal of the cover 2). Some of them are subtasks that are ready described on the same breakdown indenture level (eg the removal or the installation of the equipment 401). Both are possible, however, it is not necessary to describe the removal of the cover 2 again within the repair procedure. This is already done at the breakdown level of the cover 2. Therefore, it is recommended to use a reference to this existing task. Never use copies of existing task structures. It is possible that special supporting tasks may be referenced frequently within a project. Therefore, for updating reasons, the original information should be documented only once within the LSA database. In case of modifications of the supporting task, the change needs only to be performed once. All references would be updated automatically.
Repair procedure  
(for equipment 401)

<table>
<thead>
<tr>
<th>Subtask 1</th>
<th>Reference on: Fault location procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtask 2</td>
<td>Remove cover 1 (opening 4 quick fasteners)</td>
</tr>
<tr>
<td>Subtask 3</td>
<td>Reference on: Remove cover 2</td>
</tr>
<tr>
<td></td>
<td>Working step 1</td>
</tr>
<tr>
<td></td>
<td>Working step 2</td>
</tr>
<tr>
<td></td>
<td>Working step 3</td>
</tr>
<tr>
<td>Subtask 4</td>
<td>Reference on: Remove equipment 401</td>
</tr>
<tr>
<td></td>
<td>Working step 1</td>
</tr>
<tr>
<td></td>
<td>Working step 2</td>
</tr>
<tr>
<td></td>
<td>Working step 3</td>
</tr>
<tr>
<td></td>
<td>Working step 4</td>
</tr>
<tr>
<td>Subtask 5</td>
<td>Reference on: Disassemble equipment 401</td>
</tr>
<tr>
<td></td>
<td>Working step 1</td>
</tr>
<tr>
<td></td>
<td>Working step 2</td>
</tr>
<tr>
<td>Subtask 6</td>
<td>Remove defective component 5</td>
</tr>
<tr>
<td>Subtask 7</td>
<td>Install new component 5</td>
</tr>
<tr>
<td>Subtask 8</td>
<td>Reference on: Assemble equipment 401</td>
</tr>
<tr>
<td>Subtask 9</td>
<td>Reference on: Install equipment 401</td>
</tr>
<tr>
<td>Subtask 10</td>
<td>Reference on: Test function of equipment 401</td>
</tr>
<tr>
<td>Subtask 11</td>
<td>Reference on: Install cover 2</td>
</tr>
<tr>
<td>Subtask 12</td>
<td>Install cover 1</td>
</tr>
</tbody>
</table>

ICN-B6865-S3000L0053-001-01

In Fig 5 you can see a typical arrangement of a rectifying task. There are both subtasks and references to existing supporting tasks within the sequence of the entire repair procedure. In an extreme case, a rectifying task may contain only references to existing supporting tasks.

Note
The use of references is recommended for all activities, which are always the same, independent of where and how often the IuA is installed in the product (e.g., disassemble or assemble tasks are normally always the same, when the items are removed from the product). Installation and removal tasks on the other hand can be very different, if an item is installed in several locations in the product.

5.4 Task preconditions, pre-work and post-work
For each task, a number of preconditions must be fulfilled in order to carry out the activity. Such preconditions are of special interest for technical publications. Preconditions concerning safety must be carefully documented. The following aspects should be covered:

− General preconditions:
  This area should describe the general preliminary work that must be carried out in order to achieve the conditions, so that the real task can be started.

− Safety preconditions:
  This area should cover all aspects necessary to achieve conditions in which the task can be carried out in a safe environment (warnings and cautions).

− Personnel preconditions:
  In this area, all personnel aspects should be covered. This includes the requirement of
additional abilities or certifications of the person as well as the need for special personal requirements to achieve safety regulations.

The activities that must be carried out before reaching the required preconditions should appear in the sequence of the complete maintenance task (supporting or rectifying) as corresponding subtasks.

Another aspect is the identification of pre-work and post-work activities. In extended repair or replace procedures containing a large number of subtasks, there can be a number of subtasks which are necessary for the preparation of the subject repair or replace procedure (eg gain access or failure location procedures). Such subtasks for preparation should be marked in the LSA database as pre-work. The same applies to post-work activities, which do not belong to the real repair or replace procedure, but have to be performed in order to actually close the complete maintenance task (eg cleaning of support equipment, paperwork).

5.5 Narrative description
First of all, it should be pointed out that the narrative description of a maintenance activity is primary a matter of technical documentation. For this reason, performing one and the same activity twice should be avoided. An extended description of a task within the LSA is not desirable. A short description of the working steps can be given, but the wording should be brief. The LSA needs to identify the maintenance activities and analyze the identified tasks. The use of narrative texts in the LSA as a draft for technical documentation should be verified carefully. Technical documentation normally must obey strict rules concerning language, formatting and layout.

6 Task frequency
An important piece information concerning maintenance activities is the frequency of the analyzed rectifying task. The frequency that a maintenance activity will be performed within a year is of central significance for the planning of logistic resources. The task frequency is directly linked to the frequency of the triggering event. The frequency of some events can be foreseen rather well, while others can be only estimated with the help of statistic methods.

6.1 Maintenance activities due to inherent equipment failures
Equipment failures statistically occur after a logistic time parameter called the Mean Time Between Failure (MTBF). This is a value for the average duration between two failures of equipment. With the assumption of a constant MTBF over the entire usage time and a single installation of equipment, a predicted task frequency of a rectifying task can be calculated with a simplified formula:

\[
TF_{rec} = \sum_{i=1}^{k} FMR_i \cdot \frac{AOR}{MTBF} \cdot \lambda_{corr} \cdot \lambda_{MB}
\]

Table 8 Formula symbols (task frequency rectifying tasks)

<table>
<thead>
<tr>
<th>Formula symbol</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(TF_{rec})</td>
<td>Task frequency of the rectifying task (1/year)</td>
</tr>
<tr>
<td>(AOR)</td>
<td>Annual operating requirements [operating hours/year]</td>
</tr>
<tr>
<td></td>
<td>For equipment that works in continuous operation, (AOR) is 8760 hours per year. The measurement base of the (AOR) can also be something other than operating hours (eg distances or number of cycles).</td>
</tr>
</tbody>
</table>
**Formula symbol** | **Explanation**
---|---
MTBF | Mean Time Between Failure [operating hours]. The measurement base of the MTBF also be something other than operating hours (e.g., distances or number of cycles).
FMR | Failure mode ratio
\( \lambda_{\text{corr}} \) | Correction factor for MTBF (in case of special conditions at special installation areas). This correction factor can differ at each installation location of equipment. Especially in the case of a multiple installation of equipment, this correction factor can be of special relevance.
\( \lambda_{\text{MB}} \) | Correlation factor to convert different measurement bases of AOR and MTBF (e.g., a MTBF is given not in operational hours, but in case of a vehicle in a specific distance measurement base like kilometers, miles).
i | Index for the identification of the FMR of the single failure mode within the analyzed equipment.
k | Number of different failure modes to be considered for the accumulation.

The formula above can be considered to be the simplest possibility. A more precise task frequency calculation is only possible with more complex formula taking into account other influence factors such as:

- No failure found factors (e.g., Built in Test (BIT) cannot duplicate)
- Correction factors for different environmental influences (\( \lambda_{\text{corr}} \) in the formula above)
- Correlation factors to convert different measurement bases of AOR and MTBF (\( \lambda_{\text{MB}} \) in the formula above)
- MTBF can be a function of time MTBF(t), then integral calculation has to be used
- MTBF can be of different types (e.g., predicted, allocated, measured), the one to be used should be selected carefully
- Correction because of induced failures can be required

The logistic value MTBF is of special interest when different types of equipments are analyzed. The distribution of the MTBF value over the entire life cycle of equipment can have different behavior depending on the type of the equipment. It should be pointed out that MTBF values can change over the life cycle of equipment. Mechanical equipment with obsolescence has behavior that is different than for, for example, electronic equipment. Such reflections will be made by reliability analysis activities. With the help of mathematical functions, different failure distribution behavior can be described in detail. In this context refer to special literature concerning reliability.

In case of multiple installation of equipment it is recommended that the task frequencies for each installation point be calculated separately. For a required accumulation of task frequencies (e.g., for maintenance tasks being performed in a special repair shop) the calculated single task frequencies can be summarized. Depending on the installation area, the frequency of the failure of equipment can be dramatically different.

### 6.2 Task frequency for supporting tasks

For supporting tasks, a task frequency cannot be calculated in the same way as for rectifying tasks. Supporting tasks are not linked directly to any event like the rectifying tasks are.
However, supporting tasks are normally "called" by rectifying tasks using the referencing method. By adding the task frequencies of the referencing rectifying tasks, a "sum up" task frequency of any supporting task can be calculated with the help of the following simplified formula:

$$TF_{\text{supp}} = \sum_{i=1}^{n} \lambda_Q \cdot TF_{\text{rec},i}$$

<table>
<thead>
<tr>
<th>Formula symbol</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TF_{\text{supp}}$</td>
<td>Task frequency of the supporting task (1/year)</td>
</tr>
<tr>
<td>$TF_{\text{rec},i}$</td>
<td>Task frequency of the referencing rectifying task (1/year)</td>
</tr>
<tr>
<td>$\lambda_Q$</td>
<td>Number of &quot;calls&quot; of the rectifying task for the supporting task to be calculated</td>
</tr>
<tr>
<td>$i$</td>
<td>Index for the referencing rectifying tasks</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of the referencing rectifying task</td>
</tr>
</tbody>
</table>

It should be noted that the task frequency of supporting tasks must be used with care since the related task requirements are already covered by the rectifying tasks themselves which reference the supporting tasks.

Care must be taken to avoid double counting any requirements.

When is the calculation of the task frequency of a supporting task meaningful?
The task frequency of a supporting task may, however, be of interest concerning statistical figures that may be used for:

− Identification of maintenance drivers to be assessed for potential improvement of design
− Quantification of maintenance activities depending on the real usage/maintenance of an item

Example: Opening and closing of covers

The remove and install tasks of a cover can be referenced many times by different rectifying tasks from many different LSA candidates. It can be important how often a cover is really removed and installed within one year because of the need for some spare parts after a certain number of removal and installation. The remove and install of the cover due to inherent failure or because of damage is normally only a small fraction of the real removal and installation activity. The dominating fraction results from the need to gain access to other equipment behind the cover.

Because of the frequent removal and installation of a cover, the design aspect needs to be user friendly (e.g., by the usage of quick release fasteners).

6.3 Maintenance activities due to damages or special events

Failures of equipment for other than inherent reasons cannot be predicted in the same way as described in Para 6.1 since the task frequency can only be estimated. If there is any experience concerning the occurrence of damages or special events, with the help of statistical investigations for example, the results can be used to get a feel for how frequent these unpredictable events occur and how frequent the corresponding maintenance activities will arise.
However difficult it is to forecast the frequency of these events, it is important to get an estimation based on the best information available. All maintenance activities, which are triggered by these events, have their corresponding requirements concerning material and personnel. To estimate the resources and the capacity utilization, these unpredictable events must be taken into account.

6.4 Scheduled maintenance activities
In the case of scheduled maintenance, no additional calculation or estimation is required because the interval of the maintenance activity is fixed by the corresponding Scheduled Maintenance Analysis (SMA) and by maintainability activities by assembling proper scheduled maintenance packages.

7 Task resources and task duration
7.1 Resources
The resources necessary to perform a maintenance tasks should be defined at a common level within the task itself. Generally, it should be possible to identify when a resource should be available within the sequence of the task. Additionally, it can be of interest for training purposes, which personnel are relevant for the usage of special support equipment. Based on the principle of the structure of tasks, which can be arranged by subtasks and references to supporting tasks, the assignment of the resources should be done accordingly. It is recommended to follow the basic principle that each resource should be linked to that activity, when the resource actually is needed. Table 10 gives an overview of possible resources and the method of linking it to the corresponding activity.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spare parts</td>
<td>Spare parts should be assigned to the subtask, where they are actually required. To get an overview of the required spare parts for the entire task, all spares from the subtasks or from the referenced supporting tasks can be summarized.</td>
</tr>
<tr>
<td>Consumables</td>
<td>Consumables should be assigned to the subtask, where they are actually required. To get an overview of the required consumables for the entire task, all consumables from the subtasks or from the referenced supporting tasks can be summarized.</td>
</tr>
</tbody>
</table>
Support equipment | Support equipment should be assigned to the subtask, where it is actually required. To get an overview of the required support equipment for the entire task, all support equipment from the subtasks or from the referenced supporting tasks can be summarized. Identical support equipment that is used in parallel should be summed up. Information can also be added about which personnel use the support equipment. In complex subtasks, more than one person could be actively using several pieces of support equipment. For training requirements, it should be possible to associate support equipment with the relevant personnel resource. The assignment of support equipment often identifies the need only. In this case, the identification initiates a process to procure or to develop corresponding support equipment based on an adequate specification. Therefore, the following cases are possible for the analyst in the MTA concerning support equipment:
- Selection of existing and proven support equipment
- Selection of a specification for support equipment
- Identification of the need to develop a new specification of completely new support equipment

Personnel | Personnel should be assigned to the subtask, where they are actually required. To get an overview of the required personnel for the entire task, all personnel from the subtasks or from the referenced supporting tasks can be summarized.

Facilities | Facilities should be assigned to the subtask, where they are actually relevant. To get an overview of the required facilities for the entire task, all facilities from the subtasks or from the referenced supporting tasks can be summarized.

Technical documentation | Technical documentation requirements (eg original manufacturer's documentation) should be assigned to the subtask, where it is actually relevant.

**Note**

Although the S3000L data model enables the material resources to be linked to both task and subtask, it is highly recommended that they are assigned to the subtask, where it is actually required. In some specific cases it can be reasonable to link a resource on task level, eg if the resource is in use during the entire task and the task contains a large number of subtasks.

On the level of the entire task, all resources from the subtasks and referenced supporting tasks should be summarized and harmonized. This summary can be presented by reports concerning the different resources (or even for all resources in one report). An example is given in Fig 6.
The summary for the install procedure from Fig 6 on task level would be the following:

**Table 11  Summary of resources**

<table>
<thead>
<tr>
<th>Resource type</th>
<th>Resource</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spare parts</td>
<td>Seal</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Washer</td>
<td>7</td>
</tr>
<tr>
<td>Consumables</td>
<td>Adhesive C</td>
<td>As required</td>
</tr>
<tr>
<td>Support equipment</td>
<td>Torque wrench</td>
<td>1</td>
</tr>
<tr>
<td>Personnel</td>
<td>Electrician</td>
<td>2</td>
</tr>
<tr>
<td>Special facilities</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>Technical documentation</td>
<td>not required</td>
<td></td>
</tr>
<tr>
<td>Mean Elapsed Time (MET)</td>
<td>6,5 minutes</td>
<td></td>
</tr>
<tr>
<td>Labor time</td>
<td>7,5 minutes</td>
<td></td>
</tr>
</tbody>
</table>

### 7.2 Resources out of references

Within the methodology of referencing on supporting tasks, it should be considered that the resources of the referenced tasks will be taken over to the referencing rectifying task. At first glance, this seems to be a proper approach. However, this approach can result in a false estimation of personnel requirements. Normally, simple working steps are performed by personnel with a lower qualification (e.g., the removal of access covers). For that reason, these personnel are normally documented within these simple supporting tasks. If the supporting task is referenced within an extended repair or replace procedure, there must be the option in this context to change the real executing personnel to another person (with another/higher qualification), who is also able to perform the referenced activity and who is involved in the complete task anyway.

Similar to personnel, the use of support equipment originating in referenced supporting tasks should be analyzed carefully. Perhaps it is possible to replace the support equipment from the original supporting task by another, which is also used within the complete rectifying task at another position and which can also fulfill the needs of the original support equipment. In this
case, it is possible to reduce the amount of different support equipment that must be used within the rectifying task.

7.3 Harmonization of support equipment and spare parts
To reduce administrative effort and to avoid needless costs, it should be ensured that all identified support equipment and spare parts will be harmonized on the product level. The list of support equipment must be analyzed whether there are options to replace several prices of equipment by combined equipment or if there are support equipment of different manufacturers with exact the same functionality. The mixture of spare parts from different manufacturers should be avoided where identical items of supply are installed in the same location.

7.4 Task location aspects
To cover all aspects concerning the location at which a task is performed three different location information types should be distinguished.

7.4.1 Location in conjunction with the product itself
Comparable with the S1000D data element Item Location Code, this aspect must be covered in S3000L, also. In previous projects the terminology "On" and "Off" tasks can be referred to frequently. This means the location of the maintenance activity with regard to the product. An "On"-task concerning location means that the activity is performed directly on the product (for example in or directly at the aircraft). For example, a removal task of a LRU is always an "On"-task, because the LRU is always removed directly from the product. The disassemble task after the removal may be performed in a special maintenance shop is a typical example for an "Off"-task concerning the location.

7.4.2 Location in conjunction with the required facility/infrastructure
Another piece of information concerning location of a task is the attribute of a required facility. Additional to the "On" and "Off" information from Para 7.4.1 the required facility is an attribute at the subtask level. The selection of a facility object as the location of a subtask does not fix automatically the location information concerning "On-" and "Off-" activities. Examples for infrastructure or facilities can be the following:

- Maintenance hangar
- Individual shops such as electronic shop or engine shop
- Movement area
- Out of area

7.4.3 Location in conjunction with the zone within the product itself
If the Product is divided into physical areas (zones) this information can be added as an additional attribute (eg a zone identifier) to the subtask. In this case, the location aspect is related to the layout of the product. The information can be used to estimate the amount of activity within each local zone of the product.

Note
Information concerning zones can also be used for the documentation of the result of a zone analysis in conjunction with an SMA. In this case, the zones will be documented in the form of non-hardware breakdown elements. The identified scheduled maintenance tasks can be documented against these breakdown elements (refer to Chap 10).

7.5 Product and system availability during maintenance performance
Further important information concerning the product or different systems within the product is the availability during maintenance activities. This information can be documented with the help of corresponding codes on different breakdown levels.
This information covers the ability to operate the product during maintenance procedures. It should be distinguished between the following situations which can be documented by using corresponding codes:

- **Product not available during maintenance**
  In this case, the product cannot be used during, for example, a repair procedure or an inspection. The product must wait until the maintenance action is finished before it is available again. A typical situation can be the removal of a defective component and the repair of the defective component (e.g., in a special maintenance shop). After successful repair, the repaired component will be re-installed into the product. During the entire repair activity on the defective component, the product was waiting.

- **Product available during maintenance with reduced capability**
  In this case, the product can be used during, for example, a repair procedure or an inspection with reduced capability.

- **Product fully available during maintenance**
  In this case, the product can be used during a repair procedure or an inspection with full capability. A typical example is the replacement of a defective component by a new one. Upon completion of the replace procedure, full capability is recovered. However, the defective component may be repaired in a special maintenance shop, but this repair does not have any influence on the availability of the product. After the successful repair, the repaired component will be stored for later use.

7.6 **Support solutions (task variants)**

Maintenance activities can differ depending upon the environmental conditions during the performance of the task. In this case, the task itself can be nearly the same, however additional support equipment may be required or the task itself is different from the original one. Reasons for variants of tasks could be:

- Performance of the task under different environmental conditions (climatic aspects)
- Performance of the task at different places
- Performance of the task for another customer with different preconditions concerning personnel and equipment of the operational site
- Permanent repair or temporary repair
- Differences from peace time to war time scenarios

The task variants should be differentiated via applicability, depending upon the project.

7.7 **Task duration**

To get an overview about the capacity utilization of personnel and support equipment, time information about the tasks is required. Within the task, it must be distinguished between the duration of a task and the working time of the personnel. The following table explains the difference between Mean Elapsed Time (MET) and labor time.

<table>
<thead>
<tr>
<th>Time Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Elapsed Time (MET)</td>
<td>Duration of the entire task. The duration should be addressed against the subtasks. The duration of the whole task can be calculated as the sum of times coming from the subtasks and from the referenced supporting tasks.</td>
</tr>
</tbody>
</table>

Table 12  MET and labor time
Time | Description
--- | ---
Labor time | Summarized duration of personnel work. The duration should be addressed against the subtasks. If more than one person is working parallel on a subtask, the labor time for each different skill level must be summarized.

For example three persons of the same skill level are working parallel within a subtask, which takes 20 minutes. In this case for this subtask the following times can be documented:

MET: 20 minutes
Labor time: 60 minutes

Concerning times/durations of maintenance tasks, it must be considered as to what kind of duration should be documented in the MTA. Taking into account the non-productive times to estimate personnel requirements is an especially important aspect. In the LSA database, normally so-called "spanner-in-hand" times are documented. This means that all preliminary work concerning preparation of the real maintenance activity (e.g. paperwork, fetching of support equipment or material) normally is not included. To take into consideration non-productive times, estimates outside the LSA should be performed.

Another aspect is the consideration of logistic delay times. Any delay caused by waiting times for logistic resources (e.g. support equipment or facilities not immediately available, waiting for personnel, waiting for materiel) is usually not documented in the LSA. To take into consideration logistic delay times, estimates outside the LSA should be performed similarly to the non-productive times described in the paragraph above.

**Note**
Waiting times caused by integral parts of a maintenance task such as the curing of materiel or the drying of paint should not be considered in the same way as non-productive times or logistic delay times. Another aspect could be the need for authorized inspection before continuing the entire maintenance task with the next step. These well predictable procedures should be documented within a maintenance task as its own subtask with its specific duration and logistic requirements.

7.8 **Parallel activities within maintenance tasks**
For the sake of completeness, a means to handle work steps that can be performed in parallel should be described. Especially for complex maintenance tasks containing many subtasks that may be performed by different personnel, the simultaneous performance of work steps should be considered. This simultaneity of activities of course has influence on MET for the complete maintenance task, as well as for the task resources concerning support equipment and personnel. This is explained by the following example:
The example above should clarify the different effects of parallel activities.

The parallel subtasks ST5 to ST3/ST4 require the support equipment SE A and SE B. For subtask ST5 the same support equipment as in subtask ST2 can be used because these activities are not performed in parallel. The consequence for the complete task is the requirement of one support equipment SE A. For support equipment SE B, we have another situation. Because of the parallel use of support equipment SE B in the partly parallel subtasks ST4 and ST5, there is a requirement of two support equipments SE B for the complete maintenance task.

A similar situation holds for the personnel. The person with the qualification mechanic basic is required in parallel for the subtasks ST3 and ST5 as well as for ST7 and ST8. This causes the requirement of an additional person P3 with the same qualification as person P1.

The following table should clarify the differences between a simple approach using a sequential methodology (one subtask after the other) or the possibility of parallel activities.

<table>
<thead>
<tr>
<th>Step by step sequence</th>
<th>Parallel activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Elapsed Time:</td>
<td></td>
</tr>
<tr>
<td>MET = 35 min</td>
<td>MET = 20 min</td>
</tr>
<tr>
<td>Support equipment requirements:</td>
<td></td>
</tr>
<tr>
<td>SE A: 1</td>
<td>SE A: 1</td>
</tr>
<tr>
<td>SE B: 1</td>
<td>SE B: 2</td>
</tr>
<tr>
<td>Personnel requirements:</td>
<td></td>
</tr>
<tr>
<td>Mechanic basic:</td>
<td></td>
</tr>
<tr>
<td>1 (P1)</td>
<td>2 (P1 and P3)</td>
</tr>
<tr>
<td>Mechanic advanced:</td>
<td></td>
</tr>
<tr>
<td>1 (P2)</td>
<td>1 (P2)</td>
</tr>
</tbody>
</table>

To document the parallel and sequential performance of subtasks, the elements subtask_timeline_event and subtask_timeline_lag contained in the Chap 19 data model can be
used to document the relationship of subtasks. This can also be used to create a draft version of a working plan by eg deriving a graphic schedule of a maintenance task from the LSA database content.

8 Training requirements

The identification of training requirements concerning maintenance activities can be derived from the maintenance tasks documented in the LSA database. In this context they need to distinguish between special training needs and training needs which are covered by the abilities which can be attained by normal professional education. However, especially in many international projects, the aspect of national situations concerning education may make it difficult to identify general training needs for each partner nation.

To support the Training Needs Analysis (TNA), there is a need to document special training requirements to perform a maintenance task. For this reason, the task should be analyzed concerning requirements on the subtask level (eg the use of a complex support equipment which requires training) as well as on the task level to judge the complexity of the task as a whole. The following aspects should be considered:

- Responsible section or department for the maintenance task
- Selected skill level of personnel performing the subtask
- Need for team training due to the complexity of the task
- Use of special support equipment which requires training
- Required special training to perform the task/subtask
- Required training methods
- Required experience to perform the task/subtask
- Complexity of the task as a whole

It should be noted that the information concerning training should be collected at the subtask level. At the subtask level it is possible to analyze each activity in detail and to assign special support equipment to the affected person performing the activity. Summarizing the results of the training information on the subtask level for the complete task is the last step.

9 Examples of assigning rectifying tasks to LSA candidates

The following examples should help to get a better understanding how to assign tasks to the corresponding LSA candidates. For that reason, a simple example of equipment should be used for the following typical cases.

![Equipment example](ICN-B6865-S3000L0056-002-01)

Fig 8 Equipment example

This equipment example contains the following components with corresponding properties:
Table 14 Breakdown of equipment example

<table>
<thead>
<tr>
<th>BEI</th>
<th>Name</th>
<th>Component type</th>
<th>Next higher BEI</th>
</tr>
</thead>
<tbody>
<tr>
<td>46-01-01</td>
<td>Equipment 401</td>
<td>Equipment (LSA candidate)</td>
<td>-</td>
</tr>
<tr>
<td>46-01-01-01</td>
<td>Device 01</td>
<td>Repairable at customer operational site</td>
<td>46-01-01</td>
</tr>
<tr>
<td>46-01-01-01-01</td>
<td>Part 01</td>
<td>Repairable at industry</td>
<td>46-01-01-01</td>
</tr>
<tr>
<td>46-01-01-01-01-02</td>
<td>Part 02</td>
<td>Non-repairable</td>
<td>46-01-01-01</td>
</tr>
<tr>
<td>46-01-01-01-01-03</td>
<td>Part 03</td>
<td>Non-repairable</td>
<td>46-01-01-01</td>
</tr>
<tr>
<td>46-01-01-01-02</td>
<td>Board 01</td>
<td>Repairable at industry</td>
<td>46-01-01</td>
</tr>
<tr>
<td>46-01-01-03-01</td>
<td>Mainboard</td>
<td>Non-repairable</td>
<td>46-01-01-01</td>
</tr>
<tr>
<td>46-01-01-04</td>
<td>Power supply</td>
<td>Repairable at customer depot site</td>
<td>46-01-01-01</td>
</tr>
<tr>
<td>46-01-01-04-01</td>
<td>Connector</td>
<td>Non-repairable</td>
<td>46-01-01-04</td>
</tr>
<tr>
<td>46-01-01-01-04-02</td>
<td>Transformer</td>
<td>Repairable at industry</td>
<td>46-01-01-04</td>
</tr>
</tbody>
</table>

![Diagram](ICN-B6865-S3000L0102-001-01)

**Fig 9 Breakdown of Equipment 401**

The following paragraphs should help to receive an impression how maintenance activities can be presented within the LSA database. From a rather simple situation like an equipment replacement to an extended one with several activities at several maintenance levels the examples should sensitize to the diversity of maintenance tasks and to the need of careful documentation of every information which is crucial to complete a maintenance activity. For each event driven situation there are several potential solutions to rectify the failure. In the examples typical possible solutions are shown. In the figures which show the representation of the LSA database content the information is reduced to the essential steps in terms of LSA.
9.1 Example 1: Major failure of equipment, non-repairable

Event:
Failure of equipment 401 which cannot be repaired by replacement of components, a complete equipment change is required

LSA candidate: Equipment 401

Complete maintenance procedure description:
1. Remove faulty equipment 401 at customer's operational site
2. Test faulty equipment 401 at maintenance shop (need for a new equipment 401 to replace the faulty equipment 401 identified). Discard/dispose faulty equipment 401 (following task)
3. Receive spare part (complete equipment 401) from customer's maintenance depot
4. Install new equipment 401 at customer's operational site (replacement of equipment 401)
   Failure rectified, product can be used again
5. Re-order new equipment 401 from industry to restock customer's maintenance depot

Rectifying task for the event:
Replace equipment 401

Following maintenance related tasks:
Discard/dispose faulty equipment

Other following activities:
Re-order new equipment 401
The following table explains the usage of formats in the schematical LSA representations.

<table>
<thead>
<tr>
<th>Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>White box, black letters</td>
<td>Breakdown element</td>
</tr>
<tr>
<td>Orange box, black letters</td>
<td>Failure at LSA candidate level</td>
</tr>
<tr>
<td>Green box, black letters</td>
<td>Rectifying task for the failure at LSA candidate level</td>
</tr>
<tr>
<td>White box, black letters, red dashed frame</td>
<td>Following task triggered by a failure at LSA candidate level</td>
</tr>
<tr>
<td>Arrows, black</td>
<td>Relations from:</td>
</tr>
<tr>
<td></td>
<td>Breakdown element to a subsequent breakdown element</td>
</tr>
<tr>
<td></td>
<td>Breakdown element to the related failure at LSA candidate level</td>
</tr>
<tr>
<td></td>
<td>Breakdown element to the rectifying task</td>
</tr>
<tr>
<td>Dashed arrows, black</td>
<td>Relations from:</td>
</tr>
<tr>
<td></td>
<td>Failure to rectifying task</td>
</tr>
<tr>
<td></td>
<td>Failure to following task</td>
</tr>
<tr>
<td>Arrows, red</td>
<td>Relation from rectifying task to supporting task (only for example 4)</td>
</tr>
</tbody>
</table>

Schematical LSA representation of rectifying and following tasks:

![Schematical LSA representation of rectifying and following tasks](image)

Fig 11  Schematical LSA representation of example 1
9.2 Example 2: Mainboard failure of equipment 401 - Option 1

**Event:**
Failure of mainboard (component itself is not repairable)

**LSA candidate:** Equipment 401

**Complete maintenance procedure description:**
1. Remove faulty equipment 401 at customer's operational site
2. Receive spare part (complete equipment 401) from customer's maintenance depot
   Install new equipment 401 at customer's operational site (replacement of equipment 401)
   → Failure rectified, product can be used again
3. Remove faulty mainboard from equipment 401
   Discard/dispose faulty mainboard
4. Receive spare part (new mainboard) from customer's maintenance depot
   Repair faulty equipment 401 by installation of new mainboard
5. Send back repaired equipment 401 to customer's maintenance depot
6. Re-order new mainboard from industry to restock customer's maintenance depot

**Rectifying task for the event:**
Replace equipment 401

**Following maintenance related tasks:**
Dispose faulty mainboard
Repair faulty equipment 401 by installation of new mainboard

**Fig 12** Mainboard failure of equipment 401 - rectified by equipment 401 replacement
Other following activities:
Re-order new equipment 401

Schematical LSA representation of rectifying and following tasks:

9.3 Example 3: Mainboard failure of equipment 401 - Option 2

Event:
Failure of mainboard (component itself is not repairable)

LSA candidate: Equipment 401
Complete maintenance procedure description:

1. Remove faulty equipment 401 at customer’s operational site
2. Remove faulty mainboard from equipment 401
   Discard/dispose faulty mainboard
3. Receive spare part (new mainboard) from customer's maintenance depot
   Repair faulty equipment 401 by installation of new mainboard
4. Re-install equipment 401
   ☐ Failure rectified, product can be used again
5. Re-order new mainboard from industry to restock customer's maintenance depot

Rectifying task for the event:
Repair equipment 401 by replacement of mainboard

Following maintenance related tasks:
Discard/dispose faulty mainboard

Other following activities:
Re-order new mainboard

Schematical LSA representation of rectifying and following tasks:

Figure 15  Schematical LSA representation of example 3
9.4 Example 4: Failure of device 01, repairable at operational site

Event:
Failure of Device 01 (component itself is repairable at operational site)

LSA candidate: Equipment 401 and device 01

Complete maintenance procedure description:

1 Remove faulty equipment 401 at customer’s operational site
2 Remove faulty device 01 from equipment 401
3 Remove faulty part 02 from device 01
Discard/dispose faulty part 02
4 Receive spare part (new part 02) from customer's maintenance depot
Repair faulty device 01 by installation of new part 02
5 Re-install device 01 into equipment 401
6 Re-install equipment 401
Failure rectified, product can be used again
7 Re-order new part 02 from industry to restock customer's maintenance depot

Rectifying tasks for the event:
Repair equipment 401 by re-installation of repaired device 01
Repair device 01 by replacement of part 02

Following maintenance related tasks:
Discard/Dispose faulty part 02

Other following activities:

Re-order new part 02

Schematical LSA representation of rectifying and following tasks:

Note
Example 4 demonstrates a possible solution for a more complex situation as in the examples before. On closer examination the example describes a classical problem how to document rectifying maintenance tasks. Device 01 and equipment 401 are both repairable. At the same time device 01 is a component of equipment 401. That means, both could serve as typical LSA candidates which can be repaired by the replacement of components. This situation can be described as "full LSA candidate contains another full LSA candidate". The way to handle this situation as described in example 4 is one possible solution. The failure is documented against equipment 401 (in this case, testability and detectability properties must allow the failure detection and localization on the level of equipment 401). Consequently, the rectifying task is also documented against the equipment 401 (here Repair faulty equipment 401 by replacement of part 02 within device 01 and the subsequent re-install of the repaired device 01). The detailed activity within device 01 (remove, disassemble, assemble, re-install), can be documented as supporting tasks within device 01 as a partial LSA candidate (the rectifying task will refer to these supporting tasks). The exchange of part 02 will be documented as a work step within the rectifying task to trigger the need for the spare part itself (in this case part 02).
Depending on the situation whether the entire product is waiting for the re-installation of a faulty equipment/component after any repair activity, the representation within the maintenance tasks in the LSA database is of different character. The need for spare parts is influenced on this decision (wait for repair or not). In general two main situations can occur:

Table 16  Different spare part triggers

<table>
<thead>
<tr>
<th>Product waiting situation</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product is waiting for repair of equipment</td>
<td>The entire equipment is not needed as a spare part, because the same equipment will be re-installed (after a successful repair) into the product</td>
</tr>
<tr>
<td>Product is not waiting for repair of equipment</td>
<td>The entire equipment is needed as a spare part, because a new equipment will be installed to repair the product and to be able to reuse the product as soon as possible. The repair of the equipment will be done later, when the product is already in use again.</td>
</tr>
</tbody>
</table>

9.5 Example 5: Complex maintenance procedure on several levels

Event:
Failure of power supply (component itself is repairable at operational site by replacement of part, part itself is repairable at industry site)
LSA candidate: Equipment 401 and power supply

Complete maintenance procedure description:
1. Remove faulty equipment 401 at customer's operational site
2. Remove faulty power supply from equipment 401 at customer's operational site
3. Receive spare part (new power supply) from customer's maintenance depot
4. Install new power supply into equipment 401
5. Re-install equipment 401
   ☑ Failure rectified, product can be used again
6. Remove faulty transformer from power supply
7. Receive spare part (new transformer) from customer's maintenance depot
8. Repair faulty power supply by installation of new transformer
9. Send back repaired power supply to restock customer's maintenance depot
10. Send back faulty transformer to industry
11. Repair faulty transformer at industry site
12. Industry to send back repaired transformer to restock customer's maintenance depot

Rectifying tasks for the event:
Repair equipment 401 by replacement of power supply

Following maintenance related tasks:
Repair power supply by replacement of transformer at operational site
Repair transformer at industry site

Other following activities:
Send back repaired power supply from operational site to customer's maintenance depot
Send back repaired transformer from industry to customer's maintenance depot
Schematical LSA representation of rectifying and following tasks:

- **Equipment 401**
  - **46-01-01**
    - **Failure of power supply**
    - **Repair faulty equipment 401 by replacement of power supply**
  - **Power supply 46-01-01-04**
    - **Failure of transformer**
    - **Repair power supply by replacement of transformer**
    - **Transformer 46-01-01-04-02**
    - **Repair transformer at industry**

Fig 19  Schematical LSA representation of example 5

ICN-B6865-S3000L0107-001-01
# Chapter 13

## Software support analysis

### Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software support analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1. General</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>2. Software support analysis in the different project phases</td>
<td>4</td>
</tr>
<tr>
<td>3. Breakdown concept</td>
<td>5</td>
</tr>
<tr>
<td>3.1 Functional and physical breakdown principles</td>
<td>5</td>
</tr>
<tr>
<td>3.2 Physical software categories</td>
<td>5</td>
</tr>
<tr>
<td>3.2.1 Field-loadable software</td>
<td>6</td>
</tr>
<tr>
<td>3.2.2 Shop-loadable software</td>
<td>6</td>
</tr>
<tr>
<td>3.2.3 Resident software/firmware</td>
<td>6</td>
</tr>
<tr>
<td>3.3 LRU and SRU aspects</td>
<td>6</td>
</tr>
<tr>
<td>3.4 Functional software categories</td>
<td>6</td>
</tr>
<tr>
<td>3.5 Data</td>
<td>7</td>
</tr>
<tr>
<td>4. Software support analysis</td>
<td>7</td>
</tr>
<tr>
<td>4.1 SSA process – commonality with the LSA business process</td>
<td>7</td>
</tr>
<tr>
<td>4.2 SSA candidate selection</td>
<td>8</td>
</tr>
<tr>
<td>4.2.1 Selection of physical candidates (operational/maintenance aspects)</td>
<td>8</td>
</tr>
<tr>
<td>4.2.2 Selection of functional candidates</td>
<td>8</td>
</tr>
<tr>
<td>4.3 Maintenance relevant events for software (support initiators)</td>
<td>9</td>
</tr>
<tr>
<td>4.3.1 Operational events</td>
<td>9</td>
</tr>
<tr>
<td>4.3.2 Technical events</td>
<td>9</td>
</tr>
<tr>
<td>4.3.3 Software improvement requirements</td>
<td>9</td>
</tr>
<tr>
<td>4.3.4 Software failures</td>
<td>10</td>
</tr>
<tr>
<td>4.4 FMECA/FMEA aspects</td>
<td>11</td>
</tr>
<tr>
<td>4.5 SMA for software</td>
<td>11</td>
</tr>
<tr>
<td>4.6 LORA aspects</td>
<td>11</td>
</tr>
<tr>
<td>4.7 Software support tasks</td>
<td>12</td>
</tr>
<tr>
<td>5. Software support concept framework</td>
<td>12</td>
</tr>
<tr>
<td>5.1 Software support profile</td>
<td>13</td>
</tr>
<tr>
<td>5.1.1 Software support levels</td>
<td>13</td>
</tr>
<tr>
<td>5.1.2 Software support roles</td>
<td>14</td>
</tr>
<tr>
<td>5.1.3 Software support activities (scenarios)</td>
<td>15</td>
</tr>
<tr>
<td>5.2 Support classes</td>
<td>15</td>
</tr>
<tr>
<td>5.2.1 Processes</td>
<td>15</td>
</tr>
<tr>
<td>5.2.2 Product</td>
<td>15</td>
</tr>
<tr>
<td>5.2.3 Environment</td>
<td>15</td>
</tr>
<tr>
<td>5.3 Support functions and software related support tasks</td>
<td>16</td>
</tr>
<tr>
<td>5.3.1 Operational servicing support</td>
<td>16</td>
</tr>
<tr>
<td>5.3.2 Management support</td>
<td>19</td>
</tr>
<tr>
<td>5.3.3 Software modification tasks</td>
<td>20</td>
</tr>
<tr>
<td>6. Supportability factors</td>
<td>22</td>
</tr>
<tr>
<td>6.1 Compatibility matrix</td>
<td>22</td>
</tr>
<tr>
<td>6.2 Deployment</td>
<td>22</td>
</tr>
<tr>
<td>6.3 Documentation</td>
<td>22</td>
</tr>
<tr>
<td>6.4 Easy loading/unloading and installation</td>
<td>23</td>
</tr>
</tbody>
</table>

Applicable to: All
6.5 Easy transportation on hardware carriers ...................................................... 23
6.6 Expansion capability ...................................................................................... 23
6.7 Modularity ......................................................................................................... 24
6.8 Modification frequency .................................................................................. 24
6.9 Recovery ........................................................................................................... 24
6.10 Safety integrity ............................................................................................... 24
6.11 Security ........................................................................................................... 25
6.12 Size .................................................................................................................. 25
6.13 Skills ................................................................................................................ 25
6.14 Software distribution ..................................................................................... 26
6.15 Standardization .............................................................................................. 26
6.16 Technology ..................................................................................................... 26

List of tables
1 References ........................................................................................................... 2
2 Failure classes .................................................................................................... 10
3 Software support levels .................................................................................... 14
4 Tasks for disposal of new functionality and/or ability ....................................... 17
5 Tasks for recovery of a system and for problem reporting in case of failures .... 18
6 Organizational tasks .......................................................................................... 19
7 Maintenance support tasks ............................................................................... 19
8 Software modification tasks ............................................................................... 21

List of figures
1 Software support analysis in the different project phases ..................................... 4
2 Aspects of a SSC .................................................................................................. 13

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 3</td>
<td>LSA Business process</td>
</tr>
<tr>
<td>ASD S4000M</td>
<td>International procedure handbook for the development of scheduled maintenance programs for military aircrafts</td>
</tr>
<tr>
<td>MSG-3</td>
<td>ATA MSG 3 - Operator/Manufacturer Scheduled Maintenance Development</td>
</tr>
</tbody>
</table>
| RCM | Examples include
MIL-STD-1843 (USAF) - Reliability-centered maintenance for aircraft, engines and equipment
MIL-STD-2173 (AS) - Reliability-centered maintenance for naval aircraft, weapon systems and support equipment
UK MoD DefStan 02-45 - Requirements for the application of reliability-centred maintenance - techniques to HM ships, submarines, royal fleet auxiliaries and other naval auxiliary vessels |
1 General

1.1 Introduction

In modern products, software aspects are of increasing importance. More and more functionalities are supported or realized by complex software packages. For hardware components, concepts and processes are established to guarantee a proper supportability of the system during the entire life cycle. The appropriate tool to achieve this goal for hardware is the LSA process. The same requirement applies for software. Hardware and software are of equal importance for the proper function of a product. For this reason, an analysis methodology called Software Support Analysis (SSA) should be applied in order to develop an adequate Software Support Concept (SSC).

An SSC should take into account all activities in order to allow a continued usage of software within a product. To find a common understanding of the handling of software support, a standardized concept for software support documentation should be part of the LSA plan harmonized between the contractor and the customer.

Similar to the logistic analysis tasks for hardware, software must be analyzed concerning its operational and maintenance requirements. In this area, special operational aspects are of emphasized interest for the end user of a product because this particularly influences the day-to-day business. Aspects covering software modification, such as bug-fixing and software improvement (eg software upgrades), must also be considered.

1.2 Objective

This chapter provides the logistic analyst with guidelines for how to handle the specific requirements concerning software in the environment of maintenance and operation. The interrelation between software and hardware should be defined clearly and the method of integrating software aspects into the overall LSA process should be explained.

For software itself, a clear distinction between operational/maintenance aspects and real software modification should be established. For example, operational aspects might include the simple act of loading working software or required data to equipment, whereas software modification covers aspects that deal with the change of the source code to fix a problem, improve the performance of a software package, or adapt to changes in procedures, data, or systems that affect the software performance.

1.3 Scope

This chapter is directed to ILS managers and/or LSA analysts for both contractors and customers who must analyze systems containing software. With the means of LSA and SSA, an appropriate support concept for such equipment can be established. With the help of the various analysis results it can be assured that the customer’s needs concerning supportability, readiness and operability of hardware that contains software can be achieved.
2 Software support analysis in the different project phases

Similar to the application of LSA, Software Support Analysis (SSA)\(^1\) is an analysis activity, which should be applied throughout the entire life cycle of a product containing software. In the different life cycle phases various analysis tasks must be performed. Depending on the phase, some activities increase in importance, others decrease. For example in the design and development phase, the application of SSA is of high importance to influence software architecture and software design to consider the requirements of supportability in case of later needs to load, service, or modify the software packages. In an early concept phase, SSA should help to identify basic software support requirements. During the in-service phase, similar aspects as for hardware are valid. Technical modifications will force the analyst to repeat detailed considerations concerning the logistic concepts for the affected hardware/software. For this reason, an iterative SSA/ILS process will occur in different depths, depending on the extent of the technical modifications. The activities to be performed here are basically the same as in the design and development phase and in the production and introduction phase, but normally to a smaller extent.

<table>
<thead>
<tr>
<th>Early phases</th>
<th>Design &amp; Development phase</th>
<th>Production and introduction phase</th>
<th>In-service phase</th>
<th>Waste disposal phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conception phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk reduction phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First logistic analysis tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis for identification of</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>general SSA needs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comparative analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic Software Support</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concept (SSC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Iterative SSA / ILS process**

**Detailed software analysis tasks:**
- Configuration assessment
- Software Support Analysis (SSA)
- Level of Repair Analysis (LORA) for software
- Maintenance Task Analysis (MTA) for software related tasks
- Software delivery
- Software installation and configuration

**Periodic assessment**

**Fig 1 Software support analysis in the different project phases**

In the disposal phase, it must be ensured that during the disposal of hardware that the corresponding software and/or data are handled carefully. The handling of data can be of high interest. Sensitive data must be erased from scrapped hardware (eg in a military environment). For archiving purposes, existing data must be preserved from disposed hardware components.

---

\(^1\) Methodology to analyze software with the purpose of supplying the Customer with all necessary information to establish a cost effective SSC. This includes all equipment, tools, personnel, documentation, infrastructure and required skills and training.
If existing data is needed for further operation of newly implemented systems, the possibilities of data migration must be analyzed and concepts for data migration must be established.

3 Breakdown concept

Chapter 3 provides guidance on the development and assignment of breakdown element identifiers for hardware and software. Examples of physical and function breakdowns, to include software, are provided in Chap. 3. This guidance and the examples should be followed in developing the breakdown relationships of software within the context of the hardware.

For software, it is possible to obtain a breakdown element identifier similar to that of hardware by following the concepts of physical or functional breakdowns. However, due to different characteristics, it can be difficult to associate software within the traditional physical/functional breakdown. This is especially true for advanced systems (e.g., modular avionics), where the physical location of a specific software can be vague. Often, software is not a clearly defined physical entity; therefore, it can be difficult to identify the appropriate indenture level for the software item within a physical or functional breakdown. Even in the case of a clear assignment of software to hardware, the question could arise as to where the software should be allocated:

- Is it part of the hardware element, where the software is loaded?
- Is it part of the hardware element, where the software is executed?
- Is it part of the hardware element, where it physically resides?

The breakdown, in this context, is different from the traditional physical/functional hardware breakdown and it is important to establish a consistent approach to the breakdown of software in relation to the hardware. The physical breakdown should be used to identify software within a system breakdown for operational purposes, and provide the necessary software view for traditional system maintenance purposes. The functional breakdown is a software engineering view that facilitates how software will be modified and integrated, and provides the relationships between the different software elements that must be considered when a redesign is performed.

3.1 Functional and physical breakdown principles

Software items identified within a physical breakdown are normally elements that can be handled by the operator. This is the software that must be considered for operational aspects such as, for example, loading/unloading. The software can reside on different indenture levels of the system breakdown.

It must be noted here that the allocation of physical software to a specific indenture level is actually driven by the need to maintain an adequate configuration control of the system and also to highlight the logistics impact that such software might have on standard operation and maintenance. Though traditionally software was always assigned “below” a hardware element, new configuration management issues and approaches have changed this concept to the extent that within the LSA, software can be its own replaceable unit (LRU or SRU) or a part/component by its own right. This is further explained in Para 3.3.

Software items identified within a functional breakdown are elements that are used to describe the functional/design aspects that are important for the software designers, especially because such a breakdown usually indicates the need for integration with other software. This kind of breakdown must be considered for software modification support. Thus, it is important that the functional breakdown follows as close as possible the software design.

3.2 Physical software categories

There are basically three categories of physical software: Field Loadable Software (FLS), Shop-Loadable Software (SLS) and resident software (also called firmware). The basic concept for the three categories is established in the configuration management principles that control the configuration of a specific system.
3.2.1 Field-loadable software

FLS is any software that can be installed to one or several pieces of equipment on a system/product without the need to dismount the equipment from its existing installation location. FLS is considered an item in its own right within the breakdown and therefore, a modification of FLS results in a change of the FLS part number and a change of the configuration of the system where it resides. However, FLS loading does not cause a change in hardware part number. The reason for this is, obviously, that software that is loaded without requiring disassembling hardware from a system should not force such disassembly merely to tag the hardware with a new part number due to a new software part being installed. The system configuration is affected because the function of the system changes when using a different piece of software.

3.2.2 Shop-loadable software

Shop-loadable Software (SLS) is any software that can be loaded into a LRU, but requires that the LRU to be dismounted from its installation in the system where it is located. A replacement of any hardware component is not required. A modification of the SLS not only changes the part number of the SLS, but also the part number of the LRU where it is loaded. This is the same as when a SRU is replaced by a different one, so as to maintain the form, fit and function principle that drives configuration management (when replacing software by different software, the function is changed). Note that the system configuration is affected, but in this case, only due to the change in the hardware part number.

3.2.3 Resident software/firmware

Resident software (also called firmware) is any software that can be loaded into a LRU or SRU but requires that LRU/SRU to be dismounted from its installation on the operational system and requires the replacement of a component. Although such resident software may have its own part number when it is installed on a component or loaded onto a SRU, the part number is changed for the component or SRU where it is installed. This also entails a change to the part number of the next higher assembly.

3.3 LRU and SRU aspects

From the previous section, it becomes evident that the easiest way to handle software within an LSA system breakdown is to identify such software as LRU, SRU or part/component, depending on the software category. This not only simplifies the overall system configuration management, but it also clarifies how to handle the system configuration when using FLS and there are no tags on the target hardware regarding the software it contains – this becomes a non-issue when the configuration is handled at the next higher level.

Similarly, a loading task due to a hardware repair or software update is quite easy to integrate into the overall maintenance tasks because a specific (software) SRU or LRU is affected. Thus, in principle, it is not different than performing a task on a hardware SRU/LRU and subsequently performing additional tasks at the next higher breakdown level. Therefore, loading software after a hardware repair is very similar to, for example, replacing a consumable.

3.4 Functional software categories

The functional software breakdown, as indicated above, should follow the software design requirements, as it is mainly oriented towards software modification. Thus, the classic Computer Software Configuration Item, Computer Software Configuration and unit level views may be used but, contrary to “traditional” software engineering, it is also necessary to provide the higher levels of abstraction (subsystem and system level), as this indicates interdependencies and need for integration, including the need for system or subsystem level integration rigs. In this context, software assembly items within a functional breakdown can support the grouping of functionality that is documented as a whole (same principle as for artificial assemblies within physical breakdowns). Such artificial items can be used to provide a helpful additional level of abstraction without changing the overall structure and functionality of the software design.
It should be highlighted here that the use of a breakdown element revision identifier is also recommended, but only in the event that a major version of the software is generated that significantly changes the software structure, functionality or supportability elements (e.g., change of programming language of a specific module) that might entail a modification of the support tasks and/or support infrastructure. It does not provide any added value to perform the full analysis simply because there is a change in an algorithm, but it might be essential to perform it if, for example, the software risks class changes.

3.5 Data
Data can be handled similar to software, considering it from both the physical and functional point of view, with the peculiarity that the physical data can be loaded/unloaded (while executables are usually not unloaded, only erased) but also prepared. This is different from the pure software aspect, since software modification is a design activity, while a data preparation can be considered an operational activity. The border, however, is not as clear-cut as it looks, and there might be uncertain cases.

However, in general, data would be included in the LSA database in the same way as software, including preparation if applicable, but would also be included in the functional breakdown in order to ensure the configuration and dependencies. For example, a software modification might entail a modification in the mission preparation systems.

Whether data should be considered software or a special category within the LSA database is currently open to debate and should be previously agreed upon. Though standards such as RTCA/DO-178B include data under the software heading, there are merits to consider it a separate element for logistics purposes. As of today, the software supportability field has not reached an agreement on this subject and this particular aspect should be considered before making any logistics decision.

4 Software support analysis
Software Support Analysis (SSA) is a consistent methodology to guarantee proper software supportability throughout the requirements, specification and design phases in order to define the most cost-effective support concept that meets the operational and software modification requirements. It must be ensured that the necessary support infrastructure is established in time before the product enters into the in-service phase. The main goals of a SSA are the establishment of supportability requirements concerning software in the early program phases and the influence of the software design to ensure supportability for both, system operation and later modification processes. The outline for a stand-alone software supportability program can be found in SAE JA1004, Software Supportability Program.

4.1 SSA process – commonality with the LSA business process
In general, the SSA process should cover the logistic aspects as already described in chapter 3 for the LSA process. However, within this chapter, all special aspects concerning software are covered to ensure a proper supportability analysis process for software as well as for hardware. Additionally, the interconnection aspects between hardware and software are identified.

For a product that contains software as a basic element of the complete system, a number of software support activities must be applied depending on the situations, which are the trigger for the support. In the area of software support activities, two main fields of analysis are identified:

- Operational/maintenance aspects (e.g., loading of data or software to the corresponding hardware, system recovery, data transport and archiving)
- Software design aspects (e.g., real modification of software for bug-fixing or improvement purposes)

When carrying out SSA, it is necessary to consider the relationship and differences between the above task categories, and the representation of software items in the functional and physical system breakdowns. One basic aspect is that support activities that do not involve modification
should be analyzed against physical breakdown items. It should be pointed out that the analysis results concerning operational aspects should be recorded in the LSA database. Here you can document tasks such as the loading of data and software against the carrying hardware element within the breakdown. For example, the change of equipment can require that software and/or data must be reloaded to the new equipment and configured after or before installation into the system. The usage of hardware for special purposes can require the loading of a special software or data package to support this kind of operation. Nothing will be modified within the software itself.

The software modification itself is another area in which the software source code is normally influenced directly and changes in the source code are performed. The analysis of this kind of activity is software design related. The requirements for these activities are different from the operational aspects concerning software. For this reason, the documentation of these activities should be separated from the operational aspects. The breakdown elements, to which these activities can be addressed, should be functional software breakdown elements. Software packages that should be analyzed concerning their requirements for real software modification tasks can be treated as SSA candidates similar to LSA candidates for hardware. Aspects for the selection of SSA candidates will follow in the next chapters.

4.2 SSA candidate selection

In general, SSA candidates are elements of either physical or functional breakdowns that are subject to any kind of SSA. The candidates should be selected carefully concerning operability or supportability significance. Taking into account that software support activities are divided in operational/maintenance and software modification activities, candidate selection should be carried out within these areas separately.

4.2.1 Selection of physical candidates (operational/maintenance aspects)

A SSA candidate identified in a physical breakdown should take into consideration all software items that can be loaded and/or installed separately by the operator. The candidate itself can be the hardware item that is the carrier of the software. For distributed software, it can be a subsystem or a system within the physical breakdown. Loading and/or installation tasks are typically documented against the corresponding hardware. Other operational tasks can also be documented against a software breakdown element (e.g. transportation or distribution of software).

A special area is the handling of data. Data is normally of electronic nature and is handled similar to the corresponding software. Therefore, the support of software and dependent data can normally be analyzed together (although not in the same way). Special support aspects for data concerning the selection as a SSA candidate are the following:

- Is a data preparation process required that needs special software or hardware?
- Is a data validation process required?
- Are special transmission media or networks required?
- Are there special security aspects to be considered for data loading?
- Are there special compatibility requirements with the existing executable software?
- Are there special requirements concerning size and format (databases, file formats)?

4.2.2 Selection of functional candidates

A SSA candidate identified in a functional breakdown should take into consideration all software items that are subject to software modification activities. Each software item should be analyzed concerning the following questions in order to decide whether it is subject to a software modification activity:

- Are there any support initiators to be expected for the software items that require a modification of the software source code?
- Does the software item in the breakdown include all software within the system or within the equipment?
− Does the software item use a separate design?
− Does the software item require special hardware and/or software tools for development?
− Does the software item contain proprietary software such as run-time libraries or COTS (Commercial Off-The-Shelf) elements?
− Are there different versions of the same software item for usage on different platforms?
− Are there deviations of the software items to the general design environment?

4.3 Maintenance relevant events for software (support initiators)

Similar to the methodology of LSA for hardware, the starting point of documenting the maintenance relevant aspects should be the identification of any event that initiates maintenance activities. These events can also be described as software support initiators. The event driven methodology should also be applied to SSA, however, the events concerning software are somewhat different from the events for hardware. They can be grouped according to the implication of:

− Operational events
− Technical events
− Software improvement requirements
− Software failures

4.3.1 Operational events

This type of event is related to an operational requirement. Operational events should be documented within the related hardware item. The same applies to the corresponding operational task. Typically the following aspects should be covered:

− Hardware maintenance/repair of computing hardware

  After hardware maintenance, different types of tasks may be necessary, eg reloading of data and/or software, installation and configuration of software packages on a new installed equipment.

− Usage (mission) preparation

  The preparation for special usage of a system may require (besides the installation of additional hardware) the installation of supporting software packages or the loading of special mission data.

− Post mission requirements

  After a special usage of a system, data created during the mission must be unloaded or archived. Software packages that were required for the special mission must be de-installed and the system must be reconfigured to a standard configuration.

4.3.2 Technical events

Besides the hardware maintenance, other technical events can trigger software support activities. These events normally require adaption of the existing software packages to the modified environmental preconditions. Typically the following aspects should be covered:

− Changes of the parent hardware
− Changes of the parent software (eg upgrade of operating systems, of corresponding database systems or of firmware)
− Changes in technology of inter-operating systems
− Changes in technology of network interconnectivity

4.3.3 Software improvement requirements

Software should be the subject of a permanent improvement process. Normally, software release changes take place in particular intervals dependent upon the complexity and size of the software package. The primary inputs for new releases are (besides the fixing of real
4.3.4 Software failures

System failures initiated by software are, of course, main support initiators. The reaction to failures depends on the severity of the failure. Not every failure caused by software will initiate a software modification activity.

Considering software failures, it is recommended to introduce a classification concerning severity. For each failure class, a sequence of actions should be defined within a SSA. In every case, proper documentation is required to support later diagnosis.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minor</td>
<td>In general, these are failures concerning the user comfort. In many cases, these events are not real failures but only poorly designed features of the software or they are real failures that can be bypassed by other features of the software or an alternate handling. However, these minor events should be documented and collected carefully and reported to the software support organization. This information should guarantee the incorporation of improvements into a further release of the software package. It should be pointed out that especially the correction of marginal insufficiency can considerably increase user's acceptance.</td>
</tr>
<tr>
<td>Medium</td>
<td>These are events that cause decreased functionality of the system. If this type of bug occurs, the user is not able to continue the work as expected, however, the system as a whole works without severe disturbance (e.g., no shutdown will be performed). The ongoing usage of the system is possible with reduced performance or even with no restrictions. These events normally need a timely correction, and a defined workaround to deal with the failure until the final correction is made with the help of a bug fixing procedure. Normally, a software modification is necessary.</td>
</tr>
<tr>
<td>Major</td>
<td>A major software failure should be considered as an unacceptable event. The failure causes the shutdown of the entire system or the necessity to operate the system under very restricted conditions. The normal capability of the system is considerably degraded. These events need a correction within a small amount of time. In case of emergency situations, the software support organization should react as soon as possible to recover the entire system to full operation. The reaction times should be defined within service contracts between the customer/user and the contractor.</td>
</tr>
</tbody>
</table>

A software failure cannot be treated exactly like a hardware failure. In the case of a hardware failure, any relevant maintenance action such as a repair or a replace can be defined rather clearly. In case of a software failure, it is not normally possible to identify the required activity immediately. Perhaps a restart of the system is sufficient to recover the system and the failure will not occur again. On the other hand, the failure can be severe and the system performs a shutdown and should not be restarted in order to avoid the repetition of the failure and a further corruption of the entire system.

An important aspect can be the corruption of data or executable code. This event can be initiated, for example, by a virus infection or by corrupted carrying hardware (e.g., the loss of
functionality of a hard disk). In this case, there is no real inherent failure in the software source code. This event should be handled as external damage.

4.4 FMECA/FMEA aspects

The results of a technical Failure Mode, Effects and Criticality Analysis (FMECA) are relevant for software since the FMECA represents an analysis method carried out for the overall equipment/system (covering hardware and software aspects). For any identified failure mode, a decision should be made as to whether the associated functionality is dependent on or even provided completely by software. During the design/development of the software, this failure mode information can be used to design a software architecture that eliminates, or at least mitigates, the possibility of the software failing in such a manner as to cause a specific system failure. In addition, the design can be instrumented so as to ensure that any such potential software failures are detected and recorded, including their associated information, and that the system continues operation in a safe mode.

During the in-service phase, system failures due to software failures are drivers for software modification activities. It is a principle of FMECA/FMEA (Failure Mode and Effects Analysis) to group together all of the software failures that lead to one specific system failure, and perhaps even to group together the specific system failures that lead to the same software modification request. The analysis should be based on the functionality that the software performs. However, it is typically not practical to carry out the FMECA/FMEA of the single functional failure effects down to the level of specific defects in the source code.

Another consideration during the in-service phase is not only to provide the necessary information to the software engineers about a specific failure, but also to prevent unnecessary maintenance tasks due to software failures. A software built-in-test reporting a software specific failure that can be identified by the operator will reduce the number of disassemblies (as you cannot rectify a software failure by replacing the equipment) and number of hardware No Failure Found cases.

4.5 SMA for software

Scheduled maintenance analysis (SMA) such as ASD S4000M, MSG-3 or RCM is not applicable to software packages in the same way as for hardware. Software failures are the result of unintended effects of the software design. Such failures cannot be avoided by the means of SMA since they correspond to design flaws. However, potential software failures can affect safety. It could be possible that the result of SMA carried out for a piece of hardware equipment containing software identifies scheduled maintenance tasks that are of operational character concerning software. However, the analysis remains a SMA for hardware, taking into account software elements. A stand-alone software package normally will not be a candidate for SMA.

A SMA on the software itself should not be considered as performing regular maintenance tasks on the software itself, but rather as periodic assessments of it during the software life. It must be kept in mind that software to a great extent shows a “bathtub” like failure profile, with initially many failures. These bugs are fixed with successive software releases leading to a stable behavior. Later, failure rates may increase again due to the increasing degradation of the software as it is modified beyond its initial scope. A periodic assessment of the failure rate could determine whether the software is reaching its end of life due to increasing software failures or because the added functionality can no longer be properly processed by the underlying hardware, thus forcing either a software rewrite or perhaps even a complete system redesign.

4.6 LORA aspects

The identification of which support level a task will be performed can be supported by means of a Level of Repair Analysis (LORA) procedure. For operational aspects, the performance of tasks such as software or data loading can be covered within a LORA for the hardware related tasks, or at least by means of a process that is very similar to it. Note, however, that the “triggers” for such tasks may sometimes not be the traditional identified tasks, but might be due
to external factors such as a new software release, or the need for specific operational software or data loading. If these tasks occur frequently enough, the result of the LORA might actually be quite different than the one that would be obtained when associating these tasks to the corresponding hardware.

For real software modification, detailed information is required concerning equipment (software development environment concerning hardware and software developer tools) and capabilities of the personnel. Also, contractual warranty and software ownership aspects can influence the determination of an appropriate support level for software modification activities. For logistics management tasks, the process also becomes important in order to determine the best location to perform tasks such as, for example, generation of software or data media.

4.7 Software support tasks
After the identification of the relevant support initiators (events), the appropriate tasks (operational/maintenance tasks or modification tasks) must be identified. It should be pointed out that there is a flexible boundary, but it must be ensured that none of the support activities concerning software be forgotten. It doesn't matter whether a task will be defined as operational, maintenance, or modification relevant, the identification of the related resources and further task characteristics, such as duration, man power requirements, preconditions or safety conditions is the main goal.

To give guidelines for the analyst regarding which type of software support tasks can occur within the different areas of events, the following Para 5 gives a more detailed overview. Further information can also be found in SAE JA1005, Software Supportability Implementation Guide.

5 Software support concept framework
Similar to hardware, the achievement of a proper supportability framework should be one of the most important design principles for software. This requires the establishment of a SSC (in an early stage of the software design process). The earlier the design pays attention to logistic requirements, the better the supportability will be. It should be easy to modify software if change requirements are addressed from different sources. Also loading, installation and configuration procedures should be as easy as possible.

To take into account all of these aspects, a basic guideline should be given to the logistic analyst using this specification. The framework should cover all of the important areas concerning software support. This guideline should support contractual agreements between the customer and the contractor, which are agreed upon during the LSA GC, and all of the aspects should be tailored to the required extent of the Project.

There are different perspectives of a SSC, which shall be described in the following paragraphs. All of these views are interconnected and influence each other.
It depends upon the extent of a software development or procurement project, whether the process to establish a SSC can be a part of the overall LSA business process (described in Chap 3). Aspects of software support can be covered by the procedures of the general LSA business process. Relevant documents such as an Operational Requirements Document (ORD) or a Customer Requirements Document (CRD) should cover software support aspects concerning the required usage data. The LSA GC should also cover the software support aspects and the relevant and contractual agreements should be agreed upon in the documents from the LSA program plan and the GCD. The candidates for a detailed SSA can be selected within the Candidate Item List (CIL). The rules for SSA candidate selection should be fixed here, too.

As a general note it should be pointed out that in large and complex systems with a considerable amount of software, software supportability is worth being treated as its own area. A separate SSA program plan can be created and implemented, and a separate Guidance Conference (GC) for software aspects can be established. However, the interaction between hardware and software development must be considered. Software is designed to address functionality to hardware components. Therefore, both must work hand in hand and processes that guarantee proper cooperation must be harmonized, documented, reviewed and put into practice. A more detailed representation of aspects concerning a SSC framework can be found in SAE JA1006, Software Support Concept.

5.1 Software support profile

The software support profile takes into consideration the aspects of where the software support is located and which organizations and related persons are the relevant players in a software support environment. Additionally, the processes regarding how the different instances work together in an integrated way should be established and agreed upon between the customer and the contractor.

5.1.1 Software support levels

Similar to the maintenance levels within the LSA for hardware, levels of support should be addressed for software support. The support levels for software can be different from the hardware support levels and would be used as input to the LORA decision process for the software. Also, classification is similar to the maintenance levels for hardware, but some special aspects concerning software should be considered additionally. The following table gives an overview.
Table 3 Software support levels

<table>
<thead>
<tr>
<th>Support level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>This support level describes the location at which software is in operational use. Within this level, normally only simple support tasks can be carried out. These tasks are covered by the operational servicing support.</td>
</tr>
<tr>
<td>Level 2</td>
<td>An Intermediate support level can cover all support activities which are related to operational servicing support but also to management support. Especially user support in the form of help desks and hotline can also be covered by intermediate support level. An example for intermediate support level is the Service Repair Vehicle concept. The Service Repair Vehicle is able to perform, for example, communications, download of software or firmware update in a flexible way (like a quickly available help desk). It is located near to the operational area but is not part of the operational site.</td>
</tr>
<tr>
<td>Level 3</td>
<td>This support level is normally located at a central site. Activities from the management support can be covered here as well as modification support. It depends on the software packages themselves (e.g., complexity, size, modularity, programming techniques) and on the equipment (e.g., tools, personnel, facilities) of the support sites as to how far the capability to modify software packages is available.</td>
</tr>
<tr>
<td>Level 4</td>
<td>The vendor level is normally the best skilled support level for the high demanding tasks concerning software modification. In many cases, the vendor is the original developer of the software, equipped with all necessary developer tools and personnel. Often, this level is selected for modification support because the user itself does not have the required capabilities for these complex support tasks. Additionally, aspects of responsibility and liability must be considered, which can force the user to address software modification to the original vendor.</td>
</tr>
</tbody>
</table>

5.1.2 Software support roles

As a first step to clarifying software support roles, customer (buyer and users) and supplier (vendor and supporters) roles should be clarified. As a second step, the roles concerning software usage and support should be addressed to persons who are able to offer the required services. These roles can be:

- Simple user without any rights to modify or to recover any software system (not a real support role, but in general only a service receiver). The main task of a simple user should be to keep the software operationally running and report any system/software failures.
- Power user with some basic knowledge and the ability to perform a system recovery. This kind of role can be involved in the software support by performing simple operational support actions in order to keep the software running, such as installation and loading of data. Normally a power user is not involved in any software modification activities.
- System administrators are power users on a higher level. They should be able to cover all activities concerning operational support and eventually management support. Normally system administrators are not able to modify software itself, but software configuration like changing of performance parameters and loading of additional software packages or installation of updates are typical administrative tasks. System administrators are normally
concerned with the direct support for the simple users or power users in working with the software packages. Normally a system administrator is not involved in any software modification activities.

- Service provider, who is able to provide qualified operational support and additional services, such as a hotline. Service providers can be located onsite with the customer and vendor, and in some cases, are able to provide modification support.
- Vendor or original software developer, who is able to provide modification support up to the highest support level.

All these roles must be equipped with the required skills, capabilities and rights necessary to carry out their tasks. The roles, including their profile, should be documented in the SSC. The limits of competence should be clearly defined, documented and accepted by all persons affected.

5.1.3 Software support activities (scenarios)
Concerning the different support roles and support levels, a scenario should be described, indicating how the different resources at different locations work together and which processes are the basis for the cooperation. This support scenario should be agreed upon by both sides, the customer and the contractor, and should be carefully established and controlled.

5.2 Support classes
The quality aspect of a SSC is extended to all objectives involved. The implementation of quality standards to ensure the keeping of rules is recommended. In the next chapters three software support classes should be considered.

5.2.1 Processes
All processes being established should be discussed between the customer/user and the software support provider. A written documentation of how these processes should be implemented should be available. Process characteristics concerning inputs, outputs, controls and resources should be clearly addressed:

- Is there any written documentation of the process (flowcharts, descriptions, requirements, responsibilities)?
- Is an accepted standard used to describe the relevant processes (international or company internal)?
- Are there any real measurable performance control parameters available and what are the measures for performance control?
- Are inputs and outputs, procedures, requirements and an effective control mechanism defined and documented?

5.2.2 Product
Supportability aspects should be considered as early as possible in the software product characteristics. Inherent quality characteristics of software are dependent upon how well or badly the software has been designed. Requirements for a good software design include many aspects starting with the software specification and ending with a perfect user handling, supported by a proper graphical user interface. Typical quality aspects are the following:

- Modularity
- Changeability and expandability
- Simplicity and easy handling
- Stability and consistency
- Performance concerning processing speed
- Instrumentation (in the sense that the software is purposely instrumented for an understanding of how it is operating)
The better these aspects are considered in the development process, the easier it should be to handle or to support the software package. However, it should be pointed out, that even well designed software can cause a lot of problems by implementing a bad support concept.

5.2.3 Environment
Environmental aspects concerning quality start with the persons that are involved in the usage and/or support of software. To implement proper software support, personnel characteristics should be considered very carefully:

− Who is the right person for a software support job?
− What are the required skill levels and how much experience is recommended?
− How many persons are required to guarantee a trouble-free software support?
− Is the motivation of the personnel on a high level?
− What fluctuation can be expected?

But not only personnel aspects influence the quality of the environment. Very simple aspects such as the quality of physical facilities, computer workplace equipment, size of offices and quality of ventilation can ultimately affect the quality of the support services themselves.

5.3 Support functions and software related support tasks
Maintenance tasks related to software can be divided into different categories. This depends on the following aspects:

− Is the task simple software loading/unloading task or a simple software transportation task (from a device A to a device B)?
− Is it necessary to remove hardware from a system to load/unload software?
− Is the maintenance task connected to any case of failure recovery or documentation of any trouble related to software problems?
− Is any case of software modification involved?

Especially the last aspect of the list above, software modification, leads to a wide area of special support tasks, which must be carefully considered.

5.3.1 Operational servicing support
The operational servicing tasks describe all activities concerned with the day-by-day operation of the system. However, even these tasks can have a different quality and depth of impact on the usage of the system (e.g. downtime, test requirements). In general, these tasks are rather simple tasks carried out by the user on support Level 1 or Level 2, which only require the software package to be available on a compatible medium plus a corresponding installation manual. The method of installation should be considered. On a stand-alone system, a system administrator will have some other requirements that with a large computer network system. In modern network architectures, software and data installation and loading will be carried out with the help of software deployment systems supported by corresponding tools.

The following tables will give an overview of what kind of tasks can occur, including the further definition of some terms concerning the handling of software packages. These tasks will be grouped in three subcategories:

− Tasks for the disposal of new functionality and/or ability
− Tasks for recovery of a system and for problem reporting in case of failures
− Organizational tasks
<table>
<thead>
<tr>
<th>Type of task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software installation</td>
<td>Installation of a software package means the pure disposal of a possible functionality realized by the means of software by carrying out an installation routine.</td>
</tr>
<tr>
<td>Software de-installation</td>
<td>De-installation of a software package means the definite removal of a possible functionality realized by the means of software by carrying out a de-installation routine or by simple deleting of files from a storage device.</td>
</tr>
<tr>
<td>Software loading</td>
<td>Software loading should be understood as a task that is one step higher than the pure installation. In additional to the pure installation routine, which is part of the software loading, the configuration of software must be considered. In many cases, this step of software loading is more complex than the pure installation.</td>
</tr>
<tr>
<td>Software configuration</td>
<td>In this context, the term configuration means the change of internal parameters (e.g., operational parameters, granting of user rights, performance parameters, path information, database connections) for the operation of the software. This action requires no new installation of the software package itself. For the change of configuration parameters of the software, the functionality of the software package itself is sufficient. For the configuration, it is important to have sufficient access rights (e.g., administrator account for login into a software) to realize these changes.</td>
</tr>
<tr>
<td>Data loading</td>
<td>Normally, data will be loaded with the help of the existing functionality of an installed software package in order to provide special abilities to the product. The border between software and data in this case is flexible, depending on the type and format of the data that must be loaded into the existing system. Data loading tasks are also typical tasks for preparation of operation or of a mission. Data does not change the basic functionality of the installed software package. For example, the maps in a navigation system can be considered as data, the software of the navigation system uses this data, for example, for the graph of a city map. With the loading of another data set (e.g., package of another country), the functionality of the navigation system will not change (it will stay a navigation system), however the usage may possibly change, by the allocation of data.</td>
</tr>
</tbody>
</table>

A special aspect of installation and loading of any software/data is the embedded software or firmware respectively. In this case, the allocation of new functionality or abilities can be a more demanding activity. Normally, some special tools and support equipment (e.g., to gain access) are required to transfer the new software code or the new data to the storage devices of embedded software (e.g., resident storage chips on a computer main board). These requirements should be identified within the Maintenance Task Analysis (MTA) of the installation or loading task for the embedded software or firmware.

For each change of software by installation/loading it is recommended to define how proper function of the product after the change can be ensured. The necessity of a functional test and a confirmation of proper work by an adequate skilled specialist should be investigated carefully and, if required, documented (e.g., in the installation instructions).
### Table 5: Tasks for recovery of a system and for problem reporting in case of failures

<table>
<thead>
<tr>
<th>Type of task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software recovery</td>
<td>Software recovery includes all activities of basic diagnostic and simple recovery actions such as a reboot/restart of a system after a system or software shutdown. Normally, these activities can be carried out safely by low skilled personnel. Even in case of a successful recovery to full system operation, a diagnosis of the system is strongly recommended. All available data concerning the problem occurrence and the recovery actions should be carefully documented for evaluation. This might lead to a number of outcomes to immediately rectify the cause of the problem or to sustain full system operation by alternate means. If the underlying problem cannot be solved within a short time, a temporary downgrading of system capability should be considered.</td>
</tr>
</tbody>
</table>
| Recovery problem reporting      | In any event of a system failure involving software, a problem report should be generated, even after a successful simple recovery, such as restarting or reloading. This problem report should consider the following aspects:  
  - What was the system condition/configuration (software and hardware platform) at the time of failure? This information is important in the fault investigation on transitory effects, where the ability might be needed to reproduce the failure effect in an appropriate test environment or reference system.  
  - Type of software failure?  
  - Was a software recovery action successful (yes/no/partial)?  
  - Severity of failure?  
Problem reports should be communicated to the relevant personnel who would investigate the problem further to establish its priority and determine any workarounds and any needs for temporary operational limitations. |
| Post operational data extraction| Software support tasks carried out immediately after operation will generally be concerned with data extraction (for operational and/or engineering analysis) and fault investigation, and non-logistics functions such as the sanitation of classified software/data. With respect to data-related functions, a range of data administration tasks might be required (e.g., the provision and upkeep of data which is routinely needed by software a program to fulfill its operational role). Fault investigation tasks will relate to both activities on the prime system and to the deeper-level testing carried out on remote reference systems which replicate or simulate the prime system environment. |

All tasks related to recovery of a system do not normally contain any software modification activities. However, recovery of a system can require the involvement of an updated software package.

Problem reporting can be a starting point of the need for modification. Even if recovery tasks were successful, it could be necessary to modify software to rectify the failures that occurred to avoid the repetition of the failure situation.
Table 6  Organizational tasks

<table>
<thead>
<tr>
<th>Type of task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software delivery</td>
<td>In the context of operational servicing tasks software delivery covers the receipt of new software by the user from the distribution organization. Receipt implies a number of activities including both quality aspects and validation aspects:</td>
</tr>
<tr>
<td></td>
<td>− Checking condition and completeness of delivery</td>
</tr>
<tr>
<td></td>
<td>− Checking configuration status information</td>
</tr>
<tr>
<td></td>
<td>− Checking license information</td>
</tr>
<tr>
<td></td>
<td>− Checking compatibility between delivered software and the system</td>
</tr>
<tr>
<td>Operational configuration</td>
<td>This task takes into consideration the compatibility of software to special hardware. A specific software package may only work on certain variants of hardware items. On the other hand it can be possible to install/load different software packages to equipment, depending on the required functionality. The user must know what is compatible. Establishing configurations that are not tested or even not allowed must be avoided.</td>
</tr>
<tr>
<td>control</td>
<td>Within the area of operational configuration control, the compatibility of different software packages working on various pieces of hardware interconnected within a bus or network system should not be considered.</td>
</tr>
</tbody>
</table>

5.3.2 Management support
The management support tasks are tasks that cannot be considered as pure operational support. However, a modification of software is not part of these tasks either. This kind of support is located somewhere between the operational support and the real software modification tasks. The activities can be carried out across all levels of support and by users as well as by support organizations.

Table 7  Maintenance support tasks

<table>
<thead>
<tr>
<th>Type of task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem reporting</td>
<td>This type of problem reporting should cover more aspects than the recovery problem report. The implementation and upkeep of a management system for failure reporting and for corrective activities (e.g. a bug tracker system) is a central activity in this area. Within this system all actions should be monitored and organized:</td>
</tr>
<tr>
<td></td>
<td>− Prioritization related to the severity of problems</td>
</tr>
<tr>
<td></td>
<td>− Request for software modification</td>
</tr>
<tr>
<td></td>
<td>− Monitoring of software modification and bug fixing procedures</td>
</tr>
<tr>
<td></td>
<td>An additional aspect can be the continuous monitoring of system effectiveness and the development of support costs. To guarantee best software performance and supportability it can be necessary to change software because of these kinds of problems.</td>
</tr>
</tbody>
</table>
### Type of task | Description
--- | ---
System configuration control | In addition to the operational configuration control, the system configuration control is of increasing importance. The interconnection of many computer systems in large networks within a complex product requires configuration control on a higher level. Of course, as a first step, it must be ensured that no software package is installed on non-compatible hardware equipment. This is covered by operational configuration control.

As a second step, system configuration control must ensure that all components within a product that carries software/data and that are interconnected, work together properly without any problems. In modern products this can become a complex challenge. A compatibility matrix for software/hardware should be established. Only tested configurations of an assembly of hardware and software components can be operated under safe conditions.

Delivery and installation | In the context of management support tasks, these activities are carried out by the contractor or by the supporting organization. In particular, after the modification of software and the creation of a new software release, the modifying organization will be concerned with the best way of providing the new software product to the affected users, which can be a single client or in case of widespread products, thousands of clients. Packaging and distribution paths will be of special interest in the case of secret or security relevant software.

User support | User support covers a wide range of activities carried out by the supporting organization. The interactions between the supporter and the user can be:
- Support for installation, set-up or operation of software
- User help desk/hotline
- User queries, which are answered by the supporting organization by providing result reports
- User training

### 5.3.3 Software modification tasks
These tasks are related to a longer-term process of implementing changes to software. Normally these activities are carried out by a supporting organization of the contractor on a high level of support (manufacturer or vendor of software). However, modification of software can also be carried out by the customer (especially changes of lower criticality). This requires corresponding equipment as well as the appropriate skilled personnel. Software modification tasks cover the software modification activity itself (coding, change implementation, testing) as well as the related tasks, such as problem investigation and configuration control.

The reasons for the necessity to change software can be of different quality.
- Corrective changes
- Adaptive changes
- Perfective changes

The most critical reason, in general, is the occurrence of real failures, which leads to an unacceptable situation within the operation of the entire product. Such a failure will normally lead to a corrective modification of software. Also, the necessity of adaptive changes can be a critical situation, but normally this kind of change can be predicted and planned for slightly better than for a corrective activity (including emergent). The time schedule for adaptive
changes is not normally as critical as for a corrective action. However, in today’s systems it becomes more and more difficult to predict the adaption needs concerning software over a longer period.

Perfective changes are for increasing the user-friendly properties of the software or for increasing the functionality. Normally these changes are implemented through an upgrade concept. The requirements of the users will be collected during a certain time period and will be implemented within a new release of the software package.

Table 8  Software modification tasks

<table>
<thead>
<tr>
<th>Type of task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem investigation</td>
<td>The initial concern of problem investigation is usually to determine how the affected software can be further operated with the actual failure (eg with lower capability). As a next step, it must be ensured that a real software bug or a real technical problem exists. It must be avoided to trigger any real software modification because of, for example, user mishandling. After the problem has been correctly identified and any appropriate interim measures put in place, an analysis will follow on how the software can be modified so that the problem would be resolved permanently. For a proper problem investigation, the required needs should be available. This can be a simple programming environment with certain compilers and debuggers, but also a complex environment which allows simulation and failure reproduction.</td>
</tr>
<tr>
<td>Change implementation</td>
<td>The implementation of changes into software packages requires carefully considering some crucial aspects. Normally, the supporting organization should have access to a complete repository of the entire software development process from the beginning to the actual status. Coding, compilation and testing of software requires special tools, which must be available. Additional appropriate trained and experienced personnel are necessary. It should be considered that any basic change in technology can cause support problems. Examples could be obsolescence or the superseding of tools that are required for software modification or the change of basic operating systems on computers. This can have a large impact on costs or even require the upgrade of entire software packages to a new release when the former software package is not compatible with the new environment. In the SSC, the relevant standards and procedures that are used should be defined.</td>
</tr>
<tr>
<td>Change release</td>
<td>A new release of a software package marks the end of its change implementation process. The new software package containing the installable and executable files/code, which is tested and certified, can be now delivered to the customer/user. All additional information is also available for the end user, such as installation instructions and modified documentation. The supporting organization should ensure that the distribution of modified software releases is well organized. This should be a part of the overall SSC.</td>
</tr>
<tr>
<td>Type of task</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Configuration</td>
<td>Configuration control of software during modification is concerned with managing individual change development and controlling different software versions and documentation. For this reason, the usage of a software development repository is strongly recommended. The compatibility between the new software packages and the existing hardware/software configuration must be ensured. In the SSC, the relevant standards and procedures should be defined.</td>
</tr>
</tbody>
</table>

6 **Supportability factors**

To establish a proper supportability of software, the analyst must consider a range of factors that can influence software supportability. All of them must be taken into account carefully, dependent to the special project. The following list shall support the analyst, who must deal with support requirements for software. As described in the chapters above, some of the aspects are more related to operational support and some of them are related to software modification support. These supportability factors are generally either attributes of the software itself or the associated development process, or of the environment within which the software is operated and supported. For special projects, of course there can be additional aspects which are not completely covered by the following list, but it is a good starting point for any analytical activity concerning logistic aspects of software.

6.1 **Compatibility matrix**

In systems where multiple software carrying equipment work together in a network, it is important to guarantee the compatibility of the equipment. It must be ensured that different versions of software on different hardware items are able to interoperate properly in a networked system. On the other hand, the compatibility of different versions of a software package with different hardware items is also of interest. The following recommendations on where to document these compatibilities should be considered:

- Software to hardware compatibility can be documented within the LSA database within the physical breakdown containing corresponding software items.
- A compatibility matrix concerning networked equipments carrying software packages, which must document the allowed configurations of the complete system concerning compatibility of software issues, should be established outside any LSA and SSA databases.

6.2 **Deployment**

Software deployment is a crucial aspect for software supportability. Each location, in which software packages are in use, must be sufficiently supported. The linkage of the locations to support capabilities (eg installation support, user helpdesk, troubleshooting and recovery support) must be carefully considered and should be agreed upon by the customer and the support organization.

6.3 **Documentation**

In general, documentation refers to two main aspects:

- Software developer supporting documentation
- Software user supporting documentation
  - End user documentation (user handbook)
  - Administrator documentation (installation/update and configuration handbook for administrative purposes)

In order to ensure software supportability for the developers, documentation should be produced conforming to an agreed standard and it must be available to the organization.
charged with delivering software support. Any software tools used in the creation of documentation must be included in the support facility and arrangements must be defined for their life cycle support.

Concerning operational aspects, the documentation for the administrator or for any power user is of special interest. It should contain installation and configuration manuals as well as troubleshooting tips in case of some malfunctions, which can be rectified easily. In case of more severe malfunctions, the documentation must contain the relevant data for the supporting organization, which must be contacted.

6.4 Easy loading/unloading and installation
The duration of loading of software and data is of importance (this includes cost aspects). Software should be designed so that a loading procedure takes a minimum amount of time (eg a loading duration of several hours may not acceptable by the customer. For this reason, the software, as well as the loading devices, should be optimized for proper and fast loading processes. The same applies for the unloading of data or software after special missions or usages of the system. If a large amount of data must be downloaded from a system after a mission, there should be a fast and easy download procedure available.

An additional aspect of loading is the safety requirement. It must be ensured with the help of appropriate test procedures, that the loading or the installation was actually successful and the system works without any problems.

6.5 Easy transportation on hardware carriers
If it is necessary to transport software with the help of hardware devices, some aspects must be considered. Which kind of carrier will be selected is dependent upon the type, size and safety requirements of the software. Is it possible to transport the software on chip cards or USB sticks? Do you need DVD or even hard disks as a carrier for the software? What are the safety aspects (eg is it allowed to transport the software on a carrier, where the files can be modified easily)?

6.6 Expansion capability
Expansion capability is an aspect of system design. Though related to the software and sometimes influenced by the software design, it is usually a hardware characteristic that must be considered within the overall system design. It is concerned with the degree to which software may be modified without being limited by restrictions on computing resources. Examples of such constraints on computing resources are:

- Available memory
- Processor performance
- Mass storage capacity
- Input/output bandwidth
- Database capacity
- Network performance (LAN², WAN³, internet connectivity, firewall performance)

If expansion capability is not considered sufficiently, software modification will be limited considerably or modification costs will explode. Especially in systems with rather fixed hardware configurations or embedded real-time applications, expansion capability is of high importance. If

---

² Local Area Network. Computer network which covers a small geographic area (eg home network, network in an office covering, for example, a group of buildings)

³ Wide Area Network. Computer network that covers a broad area. WAN are used to interconnect several local area networks (LAN).
expansion capability is not considered, a required change of the hardware configuration can cause a large amount of additional activities and costs.

6.7 Modularity
Modularity is an attribute of the structure of a software design. Different areas of function within a software package should be structured with the help of a number of modules. The possibilities to apply modularity to a software design are determined by the choice of design method, programming tools and programming language. However, in general, the optimum approach to achieve modularity will be one that balances functional and performance requirements against the need to provide an understandable and supportable design. Insufficient modularity can result in increased modification effort and costs because of the need to implement consequential modifications in different parts of the software.

6.8 Modification frequency
Modification frequency (also called change traffic) is a measure of the rate at which software modification is necessary. Before the software is in use, estimates should be made with the help of comparison with similar applications (any data available from comparable in-service systems on change traffic and effort will be of significant value). Some calculation methods based on the characteristic properties of the software can be used for change traffic estimation. Experience during test and trial phases can also be included in the investigations. Modification frequency influences stability, software integrity and system operation. Change traffic will affect the volume of software support activity. Higher change traffic will require more software modification activity.

High modification frequency involves a higher risk of downtimes of the entire system because of implementation problems or even because of bugs from new software releases. So, a good balance between the frequency of new software releases and the needs from adaptive, perfective or corrective requirements must be found.

6.9 Recovery
In case of any system halt caused by a software failure, it must be possible to reset the system to full or at least to degraded functionality in an acceptable amount of time. This should be considered in early phases of software development in order to provide a proper instrumentation for recovery purposes. This might include:

− State of health monitoring
− Fault-tolerant handlers
− Design for debug features

This aspect is particularly important in systems deployed to space or other exposed locations. If software fails in such far away systems, it is crucial to have a potential recovery capability using the available instrumentation for operational real-time upgrade (eg download updated software and/or data).

Recovery procedures must be carefully documented and should be well known to the responsible users or administrators. Recovery procedures should be tested before a real emergency occurs. In this case it should be clear what to do and how (eg critical data can be restored from a data backup). The best backup is worth nothing if during an emergency the stored data cannot be restored because of unattended circumstances. Training for the personnel to react properly is recommended and proper function of recovery should be simulated regularly.

6.10 Safety integrity
The safety integrity required of a software package is determined by the safety criticality of the functions that it provides. The overall safety criticality of a system should be established by the application of an appropriate hazard analysis technique. The criticality of particular software items will be the consequence of the partitioning of system functions in the system design.
Designs should guarantee that software items that implement critical functions should be minimized or at least segregated from other parts of the software. System requirements should define safety criticality categories and software safety integrity levels. Constraints and requirements for software design, testing and modification will be associated with each safety integrity level.

However, safety integrity is not only a matter of software design. During transportation, installation and operation of the software, some activities can be necessary to achieve a sufficient safety level. For example, the granting of user access rights on the basis of allowed software functionalities or data areas is an important safety relevant activity, as well as the proper transportation of critical or secure data.

6.11 Security
The aspect of security primarily concerns the software itself. To meet security requirements for software usage, for example, the following methods can be applied:

- Cyclic redundancy check
- Encryption
- Digital signature
- Activation or license keys

The security classification of data, executable code and documentation can constrain software support activities. The main influence on equipment will be to impose special handling requirements. This might limit access to the software and introduce design requirements which raise the importance of specific software support tasks and equipment. The security classification of a software item will be dependent upon the application and the equipment design. Wherever possible, software should be designed in that way that highly classified software parts are physically segregated from all other software within the system. Security requirements should provide criteria for security classification of software items and should specify modification and handling constraints associated with such classifications.

Another aspect of security is of operational character. It must be ensured that a software package can operate in a secure environment. This means internal security concerning a proper user access control policy as well as external security. Intruders from outside must be held off and internal users must be limited and controlled concerning their activities (e.g., via the internet). These operational security requirements can cause a number of support activities such as:

- Network administration activities (e.g., granting access rights to specific data areas, file servers or application services)
- Firewall administration activities
- Virus detection and defense activities

6.12 Size
The size of software packages influences supportability parameters, both in terms of the level of change traffic expected and the resources required to implement modifications. The size of the software within a system is dependent upon the application and the design solution. Software requirements should state any constraints on the size of run-time software imposed by the system design. Many software support and supportability projections will be based on software size and complexity. Software development requirements should define requirements for data collection and analysis to measure software size and to verify models or estimates of supportability parameters, which are dependent on software size.

6.13 Skills
Required skill levels for personnel that are faced with software must be considered from several perspectives. The normal user must be able to handle the functions of the software in a professional manner. Operators with administrative responsibilities will require a deeper education concerning the functionality of the software in the corresponding IT environment. The
highest level of skill is required for software modification. In this case you need personnel with appropriate software engineering skills. Requirements for particular qualification are associated with the application domain and the technology or the methods used. Skill requirements will be determined by the system design, the software design and the applicable software support policy.

6.14 Software distribution
With the increasing significance of networked environments, software or rather data distribution should be considered under enhanced conditions. Software can be distributed on a variety of different media types, and distribution via the internet or other network environments, such as LAN or WAN structures, is also possible.

6.15 Standardization
Standardization should be applied to the computing environment within which the software is operated. The same holds for technologies and engineering processes used to develop the software and the associated software documentation. Standardization helps to reduce the diversity of tools and methods. This considerably reduces the effort for training of personnel and for the required equipment at the support facilities. Also, for operational aspects, standardization can be helpful (e.g., loading/unloading or backup procedures can be defined to conform to existing standards).

6.16 Technology
Technology aspects should be considered with respect to the software engineering methods and tools used in development and implementation. This includes:

- Software design methods and supporting tools
- Operating systems
- Programming languages
- Compilers and debuggers
- Software test methods and test environments (hardware and software)
- Project specific tools and techniques

Technology influences operational aspects, also. For example, the technology of storage devices within a system can influence support activities substantially or the usage of WEB based technologies can dramatically ease the maintenance of client computers in a client-server application environment.
# Chapter 14

## Life cycle cost considerations

### Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life cycle cost considerations</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 LCC objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.3 LCC typical applications</td>
<td>3</td>
</tr>
<tr>
<td>2 LCC Process</td>
<td>3</td>
</tr>
<tr>
<td>2.1 LCC Problem definition</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Alternatives definition</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Cost elements definition (Cost breakdown structure/tree)</td>
<td>4</td>
</tr>
<tr>
<td>2.4 Cost modeling</td>
<td>4</td>
</tr>
<tr>
<td>2.5 Data collection</td>
<td>4</td>
</tr>
<tr>
<td>2.6 Cost profile development</td>
<td>5</td>
</tr>
<tr>
<td>2.7 Evaluation</td>
<td>5</td>
</tr>
<tr>
<td>3 Cost breakdown</td>
<td>5</td>
</tr>
<tr>
<td>3.1 Acquisition</td>
<td>6</td>
</tr>
<tr>
<td>3.1.1 Research development test and evaluation</td>
<td>6</td>
</tr>
<tr>
<td>3.1.2 Investment or procurement</td>
<td>6</td>
</tr>
<tr>
<td>3.2 Ownership</td>
<td>6</td>
</tr>
<tr>
<td>3.2.1 Operating and support</td>
<td>6</td>
</tr>
<tr>
<td>3.2.2 Disposal or phase out</td>
<td>6</td>
</tr>
<tr>
<td>4 Entry data</td>
<td>6</td>
</tr>
<tr>
<td>4.1 Minimum data required to perform the analysis</td>
<td>6</td>
</tr>
<tr>
<td>4.2 Tailoring of data</td>
<td>7</td>
</tr>
<tr>
<td>4.3 Estimation of data</td>
<td>8</td>
</tr>
<tr>
<td>5 Outputs and reports</td>
<td>9</td>
</tr>
<tr>
<td>6 Sensitivity and trade-off analysis</td>
<td>11</td>
</tr>
<tr>
<td>6.1 Trade-off analysis</td>
<td>11</td>
</tr>
<tr>
<td>6.2 Sensitivity analysis</td>
<td>11</td>
</tr>
<tr>
<td>6.3 Simulation</td>
<td>12</td>
</tr>
<tr>
<td>7 LCC tools</td>
<td>12</td>
</tr>
</tbody>
</table>

### List of tables

<table>
<thead>
<tr>
<th>Number</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>References</td>
</tr>
</tbody>
</table>

### List of figures

<table>
<thead>
<tr>
<th>Number</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acquisition costs vs. ownership costs (iceberg effect)</td>
</tr>
<tr>
<td>2</td>
<td>Example for a LCC process logic</td>
</tr>
<tr>
<td>3</td>
<td>CBS example</td>
</tr>
<tr>
<td>4</td>
<td>LCC summary cost, yearly budget - Example</td>
</tr>
<tr>
<td>5</td>
<td>LCC summary cost, yearly budget split - Example</td>
</tr>
<tr>
<td>6</td>
<td>LCC breakdown - Example</td>
</tr>
</tbody>
</table>
1 General

1.1 Introduction
Life Cycle Cost (LCC) is the cumulative cost of a product over its life cycle from Entry into Service to disposal as determined by a process of economic analysis that allows for the assessment of the total cost of acquisition, ownership and disposal of the product.

1.2 LCC objectives
The objective of the Life Cycle Costs Analysis (LCCA) is to provide the managers with an economic enlightening of each alternative presented in order to determine the most cost-effective support. LCCA approach from a series of alternatives in order to achieve the minimum LCC that guarantee the effectiveness figures required to the product.

The following figure shows the relation between acquisition cost and support and operation cost for buying decision.

![Total LCC](image)

*Fig 1 Acquisition costs vs. ownership costs (iceberg effect)*

The costs of ownership and operation, usually exceed significantly all cost of acquisition and, therefore, the decision to purchase a system/product should be not only be influenced by the product's initial cost (acquisition cost) but also by the product's expected operating and maintenance cost over its life (ownership cost) and disposal cost.

LCCA can also be applied to evaluate the costs associated with a specific aspect, for example, the effects of different maintenance concepts/approaches to cover a specific part of a product, or to cover only selected phases of a product's life cycle. It is essential to identify what the main cost drivers are (LCC elements which have a major impact on the LCC) and what can be done
to keep them as low as possible. For this purpose, additional investigations may be offered in order to justify the significance of cost driving elements (eg so-called "sensitivity runs" and/or operational simulations).

Once the cost drivers have been identified, a decision making process could be initialized to evaluate and select the equipment and/or support solution offering the most cost effective solution to meet a defined need.

In order to support the customer in decision making, beside the purely LCCA results, a conclusive recommendation should be offered by taking into account aspects concerning standardization, potentially technological evolutions, cost trends (eg after closing of production lines), readiness drivers, risks of obsolescence, preconditions for support contracts, etc.

1.3 **LCC typical applications**

LCCA may be applied for:

- Evaluation and comparison of design alternatives. In the case of modifications or design changes to be embodied on the product, the LCCA assesses the impact on costs of the related items.
- Evaluation and comparison of alternative support concepts.
- For the same design of a mission system it is possible to consider different support concepts alternatives (eg Maintenance Level 1, Maintenance Level 2, Maintenance Level 3 and Maintenance Level 4 versus Maintenance Level 1 and Maintenance Level 4 only). It is possible to have different definitions of the support levels increasing the capabilities of a level and decreasing the capabilities of another. LCCA must be performed quite close to the support concept decision and the LORA analysis.
- Affordability studies, assessment of economic viability of projects/products. Through the LCC an estimate of the total cost of a program or project over its useful life could be obtained providing information as to whether or not a customer can afford a project or product.
- Identification of cost drivers and cost effective improvements. Having identified the cost drivers, a decision making process could be carried out to evaluate and select the equipment offering the most cost effective solution to meet a defined need.
- Evaluation and comparison of different approaches for replacement, rehabilitation/life extension or disposal of systems/products. LCCA could be used as a guide to examine the effects of different solutions upon support cost and therefore upon total costs.
- Assessment of product assurance criteria through verification tests and their trade-offs;
- Long-term budget planning and tracking. LCCA results in information and data that may help in developing budgeting or in providing inputs to the programming, planning and budgeting systems.

2 **LCC Process**

Although it could be necessary to adapt the LCC process to the system being considered, there are common steps that may be essential in all the LCC processes. The depth of the analysis for every step needs to be adapted to the requirements of each project, and therefore, some kind of tailoring will need to be agreed at the beginning of the program.

2.1 **LCC Problem definition**

1 Definition of the aims and objectives of the study
2 Definition of the boundaries and assumptions of the study
3 Cost driver identification. Refers to those items of the product that have an important contribution to the LCC
4 Constrains analysis
5 Risk identification

2.2 Alternatives definition
1 Support concept analysis
2 Current operation and standardization opportunities for support
3 First draft support alternatives identification
4 Feasibility alternatives trade off
5 Final support alternatives definition, including detailed specification (personnel skill available, storage, transport, support equipment and facilities belonging to each support alternatives)

2.3 Cost elements definition (Cost breakdown structure/tree)
1 Cost elements identification. Describing each cost element
2 Cost element key performance indicators identification
3 Cost element analysis approach. Including source data identification

2.4 Cost modeling
1 Analysis of different cost models
   Whatever cost model is to be selected, Pareto rule should be considered for selection of vital few cost contributors. The purpose of Pareto charts is to identify the vital few cost contributors so the details can be itemized and ignore the many trivial issues. Pareto rules say that 10% to 20% of the elements of a cost analysis will identify 60% to 80% of the total cost. These items are the vital few items of concern and need to be carefully considered. For example, when selecting the components with more influence on the repair cost, the key value to be considered for Pareto rule application could be "Component Unit Price/MTBF" since both high prices and low MTBF could be considered as the main cost contributors to the total repair cost. By selecting the items adding up the 80% of the total for the key value "Component Unit Price/MTBF", components with more influence on the total cost are defined.
2 Cost model evaluation.
   Experience on the existing model as well as the availability of data required for the model or consideration of efficiency analysis must be taken into account when selecting one model per each program/project. Appropriate cost model should be selected depending on the product to be analyzed and objectives of the LCCA. In some cases, existing cost models could suit needs but, in other cases, customized models could be required to archive expected objectives. The selection of the cost model should be done taking into account the benefits and also the costs of using one or another solution.
3 Cost model tool availability in the market or cost model tool design and development.

2.5 Data collection
1 Identification of data requirements for the selected model.
2 Identification of sources for data (department responsible)
3 Data collection from sources. Include preparation of requirements for equipment suppliers and subcontractors.
4 Data validation to check the quality of the data and also the coherence between different sources of a data.
2.6 Cost profile development
This is the complicated section where all the details are assembled. Of course the more thorough the collection process, the better the LCC model.

2.7 Evaluation
1. Running of the model for different alternatives
2. Sensitivity analysis performance
3. Analysis of results and decision of the best alternative. During the evaluation it is necessary to check if the results meet the criteria defined in the first step of the LCCA. If it does not satisfy the criteria, it should be modified as an alternative and the LCCA of the alternative should be evaluated in order to find the best alternative.

![Diagram of LCC process logic]

Fig 2 Example for a LCC process logic

3 Cost breakdown
The Cost Breakdown Structure (CBS) identifies all cost items or 'cost-elements' affecting the total LCC of the system/product. In order to avoid ignoring significant cost elements, it is recommended that the cost elements are defined in a systematic manner.

Cost elements depend on the system/product to be analyzed and CBS should be tailored for each application area for LCCA.

Costs can usually be divided into Non-Recurring Costs (NRC) those that occur once over a limited period of time and Recurring Costs (RC) that occur repeatedly over time according to the degree of utilization of the system/product.

The appropriate life cycle phases should be selected to suite the special needs of each specific analysis but, in a general way, the total costs of a product through its life can be broadly defined as follows:
3.1 Acquisition

3.1.1 Research development test and evaluation
Research Development Test and Evaluation - the sum of all funded costs required to bring a product’s development from inception to production.

3.1.2 Investment or procurement
Investment or procurement - the sum of in-house costs, both NRC and RC, need to transform the results of Research & Development into a fully deployed operational system.

3.2 Ownership

3.2.1 Operating and support
The costs of operation, maintenance and supply support of products and support equipment are incurred throughout the expected life of the system/product.

- Costs associated with operation
  - NRC, eg costs for initial training of staff, documentation, initial spares, equipment, facilities and special tools
  - RC, eg costs for labor, consumables, power, on-going training and upgrading
- Costs associated with preventive maintenance
  - NRC, eg costs for acquisition of test equipment and tools, initial spares and consumables, and initial training of staff and initial documentation and facilities
  - RC, eg costs for labor, spares, consumables, on-going training and documentation
  - replacement of parts with limited lifetime (may be recurring or non-recurring)
- Costs associated with corrective maintenance
  - NRC, eg costs for test equipment, tools, initial spares, initial training of staff, initial documentation and facilities
  - RC, eg costs for labor, spares and consumables, on-going training and documentation

It should be noted that operational costs (eg fuel consumption), personnel expenses, real estate costs, etc, which may arise on the customer side are either purely estimated or to be contributed by the customer.

3.2.2 Disposal or phase out
The sum of all costs required to remove the system or equipment from the inventory, and which may be offset by some residual value.

4 Entry data

4.1 Minimum data required to perform the analysis
Data required to perform the LCCA will be dependent on the complexity and detail of the cost element structure defined but, in general, the following minimum data elements are required when performing the LCCA:

- Environmental data elements. The environmental data set is divided into two categories of input variables:
  - Support constants, that are divided into:
    - Program independent factors, describing the world in which the product is operating (eg personnel costs, personnel turnover rate, inflation rate).
Contractor constants, describing the conditions under which the system is maintained (e.g., support concept limitations to be considered for the analysis).

- Deployment and operating program variables. These variables represent information about the number of products to be considered and the extent and type of operations in which they are involved (e.g., product delivery/retirement planning, product mission profile and variables).

- Product data elements. The product data elements are divided into:
  - Product data elements that describe the product's design characteristics (e.g., scheduled maintenance tasks, time to repair).
  - Product configuration items data elements, that describe the systems or parts considered for each indenture level according to the configuration tree under study (e.g., MTBF, MTBUR, time to repair, unit prices).

4.2 Tailoring of data
The selection of cost elements should be related to the complexity of the product, as well as to the cost categories of interest in accordance with the required cost breakdown structure. Differences between programs and projects require a flexible approach when defining the cost element structure and it is difficult to produce a single cost structure for all LCC applications. Nevertheless, the basic structure of the CBS should not change. Using this basic structure, the cost element structure should be expanded to levels of detail to meet the specific need being analyzed.

The following figure shows an example of a CBS expanded into four levels.
### LCC cost breakdown

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft acquisition</td>
<td>Aircraft &amp; Mission systems acquisition</td>
<td>1.1 Crew</td>
<td>3.2.1.1 Personnel</td>
</tr>
<tr>
<td>Entry into service</td>
<td>Support equipment</td>
<td>1.2 Mission personnel</td>
<td>3.2.1.2 Material</td>
</tr>
<tr>
<td></td>
<td>Spares</td>
<td>1.3 Fuel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Training</td>
<td>1.4 Consumables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Technical publication</td>
<td>1.5 Aircraft insurances</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Facilities</td>
<td>1.6 Air navigation &amp; landing fees</td>
<td></td>
</tr>
<tr>
<td>Operation &amp; Support</td>
<td>Operation</td>
<td>2.1 Scheduled maintenance</td>
<td>3.2.2 Personnel</td>
</tr>
<tr>
<td></td>
<td>Life limited parts</td>
<td>2.2 1st level maintenance</td>
<td>3.2.2.1 Material</td>
</tr>
<tr>
<td></td>
<td>Repairs</td>
<td>2.3 2nd level maintenance</td>
<td>3.2.2.2</td>
</tr>
<tr>
<td></td>
<td>GSE maintenance</td>
<td>2.4 3rd level maintenance</td>
<td>3.2.2.3</td>
</tr>
<tr>
<td></td>
<td>Training refreshment</td>
<td>2.5 Crew</td>
<td>3.2.5.1</td>
</tr>
<tr>
<td></td>
<td>Technical publication</td>
<td>2.6 Maint. technicians</td>
<td>3.2.5.2</td>
</tr>
<tr>
<td></td>
<td>Facilities</td>
<td>2.7 Other pers. training</td>
<td>3.2.5.3</td>
</tr>
<tr>
<td></td>
<td>Engagement &amp; logistic services</td>
<td>2.8 Training aids</td>
<td>3.2.5.4</td>
</tr>
<tr>
<td></td>
<td>Technical support service</td>
<td>2.9 Training aids</td>
<td>3.2.5.5</td>
</tr>
<tr>
<td>4/A/C Mods &amp; Upgrades</td>
<td>Major modification</td>
<td>3.2.8.1 Engineering services</td>
<td></td>
</tr>
<tr>
<td>5/Disposal</td>
<td>Minor modification</td>
<td>3.2.8.2 Logistic support</td>
<td></td>
</tr>
<tr>
<td>5.1 Through life disposal cost</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.2 Final disposal cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Fig 3 CBS example

#### 4.3 Estimation of data

Accuracy of input data is crucial to improve the certainty of the LCC prediction. Factual data should be available to quantify cost elements in a CBS. Each cost element may be quantified by directly applying the collected actual data to the model of LCC. If actual data are not available, the cost elements relevant to the non-available data may be estimated.

To estimate the value of data, in a particular cost data, some methods are proposed:
Analogy Method. This method draws on the relationship between current and similar previous data with adjustments made from expert judgment to reflect characteristics of the data under consideration.

Parametric method. It is based on statistical analysis of historic databases. It usually results in a cost estimating or cost factor relationship.

Engineering Method. This method consists in the application of analytic formulae that process known system/product characteristics to yield the cost value. This method is also known as "bottom-up estimate".

The selection of a cost estimating technique depends basically on the level of detailed data which is available at the time the LCCA is prepared. In general, analogy and parametric methods are most useful during the early stages of the product's life, serving as an order-of-magnitude estimate of potential costs. As the design is stabilized and more information is available, the parametric method becomes a more useful technique. Later, when there is a detailed product, engineering estimates may become a more appropriate device for estimating cost. Since analogy or parametric methods are normally less costly than the more detailed engineering estimation technique, it is important to decide whether the more detailed estimate is worth the higher cost. More detailed estimating techniques do not necessarily provide more accurate estimates.

When the characteristics of the product and its operating scenario become known the best source of information for many cost elements is the 'actual' or in-service data. In-service data should be used as it becomes available because this is often the most credible source of information.

5 Outputs and reports

The results of the life cycle costing should be documented and presented by using so many reports and graphs as required allowing users to clearly understand both the outcomes and the implications of the analysis.

The final LCCA report should contain at least the following:

- **Purpose and scope.** A statement of the LCCA objective, including a definition of the intended product use environment, operating and support scenarios; assumptions, constrains, and alternative courses of action considered in the analysis.
- **LCC model description.** A summary of the LCC model, including relevant assumptions, a depiction of the LCC breakdown structure, an explanation of the cost elements and the way in which they were estimated, and a description of the way in which cost elements were integrated.
- **LCC model analysis.** A presentation of the LCC model results, including the identification of cost drivers, the results of sensitivity analyses, and the output from any other related analysis activities.
- **Sensitivity and Trade-off analysis results**
- **Conclusions and recommendations.** A presentation of conclusions related to the objectives of the analysis, and a list of recommendations regarding the decisions that are to be based on the analysis results, as well as an identification of any need for further work or revision of the analysis.
- **Graphs.** Different graphs showing the LCC outputs. Fig 4, Fig 5 and Fig 6 show some different charts that could be obtained as result of the LCCA:
Fig 4  LCC summary cost, yearly budget - Example

Fig 5  LCC summary cost, yearly budget split - Example
6 Sensitivity and trade-off analysis

Sensitivity and Trade-off analyses are performed by changing LCC model inputs and noting the impact on the LCC results.

6.1 Trade-off analysis

Trade-off analysis deals with examinations of alternative strategies. Trade-off analysis is used to evaluate proposed alternative systems or equipment configurations in terms of estimated costs, risks, benefits and operational effectiveness. These studies are designed to assist decision makers in selecting the most desirable system/product configuration among the base line system and the alternatives evaluated in the LCCA.

During the LCCA, it is checked if the system/product meets the criteria defined at the beginning of the LCCA. If it does not satisfy the criteria, it should be modified as an alternative system and the LCC of the alternative system should be evaluated. In LCCA, the optimization process of seeking the best is applied to each alternative in accordance with the decision evaluation. In a broad sense, the optimization process generally means to find a set of parameters that minimizes the LCC of the total system/product; however, in a narrow sense; the optimization may be applied to specific activities in the LCC processes such as design optimization, maintenance optimization, etc.

6.2 Sensitivity analysis

Sensitivity analysis examines the impact of changes to cost drivers or cost with big impact in the LCC results. Varying the input parameters over a range to see the impact on cost can help highlight the major factors affecting costs.

Sensitivity analysis can be performed in two ways:

- Multiple overall values. Entries are based on a percentage of the value.
- Detailed examination of a single value. A range of values for one specific parameter is defined and also the number of iterations.
6.3 Simulation

It is very important to consider the aspect related with the simulation capabilities of the LCCA. Considering that LCCA is used to make tradeoff, LCC can be performed to simulate different support scenarios or different design solutions of a product.

Varying parameters used to calculate the cost of the whole life cycle are used during the sensitivity analysis in the same way can be simulated the impact of changes in the different characteristics of the mission system alternatives as well as the support system changes.

7 LCC tools

There are many software packages for LCCA on the market but the most appropriate tool should be selected depending on the specific system/product to be analyzed and the objectives of the LCCA.

The use of tailored LCC models could be preferred in some cases if there are requirements for adopting a range of approaches particularly to evaluate the costs associated with a specific activity, for example, the effects of different maintenance concepts, to cover a specific part of a product, or to cover only selected phase or phases of a product’s life cycle according to a supplier-user agreement.

In the same way, and taking into account that the selection of cost elements within the LCC model are related to the complexity of the product and that the cost categories of interest should be in accordance with the required CBS, standard LCC tools could not suit some specific needs and customized tools could be required for these cases.

The selected LCC tool, in order to be realistic, should:

− Represent the characteristics of the product to be analyzed, including its intended use environment, maintenance concept, operating and maintenance support scenarios as well as any existing constrains and limitations.
− Be comprehensive in order to include and highlight all factors that are relevant to LCC.
− Be simple enough to be easily understood, managed and updated in case of specific needs requiring customization of the tool.
− Concentrated on the main cost drivers that considerably influence the total LCC to avoid ignoring significant elements and also to avoid those insignificant cost elements that do not always provide additional accuracy to the final results.
− Flexible to allow the evaluation of different approaches or scenarios.

In some cases, the LCC tool may need to be specifically developed for the problem under study, while for some other cases commercially available models may be used. Knowledge of the contents and the conditions under which they apply are important in order to assure adequacy of the use of one or another tools.

Where possible it is recommended that all parties in the project use the same LCC tool.
Chapter 15

Obsolescence analysis

Table of contents

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obsolescence analysis</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>1</td>
</tr>
<tr>
<td>1 General</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>2 Obsolescence strategy</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Reactive obsolescence management</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Proactive obsolescence management</td>
<td>3</td>
</tr>
<tr>
<td>2.3 Obsolescence monitoring</td>
<td>4</td>
</tr>
<tr>
<td>2.4 Strategy tools</td>
<td>4</td>
</tr>
<tr>
<td>3 Develop an obsolescence management</td>
<td>5</td>
</tr>
<tr>
<td>3.1 Scope and content of an obsolescence management plan</td>
<td>5</td>
</tr>
<tr>
<td>4 Implementation of obsolescence solutions</td>
<td>6</td>
</tr>
<tr>
<td>4.1 Budget for obsolescence management costs</td>
<td>6</td>
</tr>
<tr>
<td>4.2 Linking obsolescence to the risk management activity</td>
<td>7</td>
</tr>
<tr>
<td>4.3 Obsolescence and life cycle costs</td>
<td>7</td>
</tr>
<tr>
<td>4.4 Ensuring industry plans and manages obsolescence</td>
<td>7</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>8</td>
</tr>
</tbody>
</table>

List of tables

1 References .................................................................................................................. 1
2 Options for reactive obsolescence management ............................................................ 3
3 Options for proactive obsolescence management .......................................................... 4

List of figures

1 Life cycle phases ........................................................................................................... 2

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

1 General

1.1 Introduction

Obsolescence occurs due to the length of time it takes to develop and field a product (Fig 1) and then the subsequent long life cycles of products. Obsolescence affects all products and systems and is not limited to hardware and components, but includes test and support.
equipment, software, tools, processes, logistic products, standards, specifications and expertise.

<table>
<thead>
<tr>
<th>Early phases</th>
<th>D&amp;D</th>
<th>Production/Implementation</th>
<th>Inservice</th>
<th>Disposal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 5 years for</td>
<td>Up to 5 years for</td>
<td>Up to 25 years for</td>
<td>30 or more years for inservice phase</td>
<td></td>
</tr>
<tr>
<td>the concept</td>
<td>the design &amp; development</td>
<td>the production phase</td>
<td>phase</td>
<td></td>
</tr>
<tr>
<td>phase</td>
<td>phase</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concept and</td>
<td>System development and</td>
<td>Production and</td>
<td>Operations and Support</td>
<td>Disposal</td>
</tr>
<tr>
<td>technology</td>
<td>demonstration</td>
<td>deployment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>development</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig 1 Life cycle phases

Obsolescence occurs for a number of reasons:

− The life spans of the components that make up the product are decreasing, especially the life cycle of electronic components (presently approximately 5 years or less per innovation cycles).
− Obsolescence occurs because the manufacturing base, subcontractors and vendors, are subject to market forces. Manufacturers can go out of business and essential parts or sub assemblies can become unavailable.
− The loss of design and technical know how can have a big impact on the supportability of long life cycle products particularly if bespoke software and components (ASICS for example) were used in the original design.
− Increasing environmental legislation regarding the use a specific chemicals (MEK for example) or materials (beryllium copper for example) has also increased the pace of obsolescence as it restricts the use of materials and this should be considered from the earliest phases of a project.
− The costs of scarce materials, production facilities and support services (software maintenance for example) can also make a product unaffordable to support.

The rate of technological innovation coupled with long utilization of systems and services mean that it is almost inevitable that obsolescence will have an impact at some time in the life cycle. Despite this inevitability, obsolescence can be managed and mitigated. The cost of mitigation however increases significantly as the system, product or service moves closer to becoming obsolete. Obsolescence management (OM) should therefore be undertaken as early as possible and as an integral part of the design, development, production and inservice support phases in order to minimize potential remedial expenditure and thereby the overall Life Cycle Cost (LCC).

1.2 Objective

The relationship between LSA and obsolescence is that both processes occur repetitively in the product life cycle and obsolescence must be an integral part of the LSA process as it impacts supportability, operability and LCC and therefore must be closely managed until the disposal phase. The objective of this chapter is to advise on the preferred practice for OM. It offers guidance on obsolescence planning to mitigate the risk of obsolescence and minimize the LCC of long life cycle products. Please note that this chapter is intended as an introduction and overview of the issues related to OM. Authoritative guidance can be found in the European standard EN 62402:2007 Obsolescence management - Application guide.
1.3 Scope
This chapter is aimed at engineering, quality, logistic, supply chain and sustainment organizations. It provides guidance on:

- Defining an obsolescence strategy
- Implementation of obsolescence solutions
- Develop an OM plan
- Linking obsolescence to the risk management activity
- Obsolescence and LCC
- Ensuring OM costs are budgeted for
- Ensuring industry plans and manages obsolescence

2 Obsolescence strategy
There are two basic strategies for managing obsolescence, reactive and proactive and these are outlined below.

2.1 Reactive obsolescence management
Reactive OM implies that there is either no specific provision for the task of managing obsolescence or for the resolution of the obsolescence issues when they occur. Each occurrence is dealt with in isolation, unless they occur concurrently, and there is no link into the risk management activity. Resolution is funded from within the budget allocated to other activities or additional funds are sought. If a reactive approach is being taken your management options are limited to:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do nothing</td>
<td>This is a valid choice. Your decision to have a reactive strategy may have been based on the extended life and very low consumption of the item. This obviates the need for additional procurement activity.</td>
</tr>
<tr>
<td>Re-design</td>
<td>Usually an expensive option but it may be unavoidable. Costs may have been mitigated by limited design transparency.</td>
</tr>
<tr>
<td>Alternative part procurement</td>
<td>This activity usually requires a parts search and also covers cannibalization of other items.</td>
</tr>
</tbody>
</table>

2.2 Proactive obsolescence management
Proactive OM implies that there is specific resource provision and a plan generated for managing obsolescence issues. It is linked into the risk management activity and has a budget allocated. It also has a stakeholder group identified for addressing issues as they arise. Monitoring of the system creates time to review resolution options and justify decisions. It may also be considered that a combination of these two approaches should be applied. This may be particularly appropriate where very large numbers of components are concerned, some of which could be identified as low risk. This option reduces the management cost of the OM task and also focuses on the high-risk items. Though there is increased risk attached to this approach which should be considered against cost and impact considerations. Care should be taken if considering reactive OM in the early phases of a project as it may not be possible to recover important information necessary to support proactive OM later in the program. It is recommended that the proactive strategy is considered initially for all programmers. If a proactive approach to OM is being taken your management options are extended accordingly to:
### Table 3  Options for proactive obsolescence management

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do nothing</td>
<td>As for reactive obsolescence management</td>
</tr>
<tr>
<td>Life time or end of life buy</td>
<td>A valid approach which can offer the best value. However care must be taken to consider the possibility of equipment life extension, changes in consumption rates and costs of ownership (eg storage, handling, exercising).</td>
</tr>
<tr>
<td>Planned upgrades</td>
<td>These offer good opportunities to address multiple issues at once, but would usually need to be coupled to obsolescence monitoring activities whether the upgrade is requirement or obsolescence driven. Care would also need to be taken to ensure sufficient spares are available between upgrades and dates and budgets strictly controlled.</td>
</tr>
<tr>
<td>Obsolescence monitoring</td>
<td>A key component of any OM strategy which if implemented correctly will at least ensure that you have time to plan and consider your solution. Once the obsolescence is noted you can then determine your preferred solution from the options available.</td>
</tr>
<tr>
<td>Design out obsolescence</td>
<td>Clearly this approach needs to be implemented in the earliest phases of a program. The intention is to design the equipment such that a sub-assembly capability is not tied to a particular solution.</td>
</tr>
</tbody>
</table>

It should be noted that mixed strategy is perfectly acceptable. This approach may be considered because proactive OM carries a cost which increases with the number of components considered and monitored. The cost needs to be weighed against the possible impact on availability and obsolescence rectification costs. The cost of OM is a driver in the choice of the strategy.

### 2.3 Obsolescence monitoring

When implementing an obsolescence monitoring solution there are various options which should be considered and some are discussed below. The use of the Design Authority (DA) to manage obsolescence is an obvious choice. The DA is best placed to develop a list of parts, identify solutions and modify the design. This is particularly so during the early phases of the project when the design may be influenced by OM considerations. An alternative solution which may be considered, particularly in the in service phase is the use of a third party contractor with a dedicated OM portfolio of skills. There are many companies which offer an OM service, each to varying depths, which can be tailored to meet the requirements of the customer. The advantages of using such a contractor may be that they have specialized skill sets in the OM area and experience of doing this work over many years. Any overhead would be spread over many other projects and therefore may offer better value. They do not have a vested interest in identifying obsolescence issues unless they are also contracted to identify solutions. A further option would be to employ a combination of the DA and third party. The DA could be employed to do the bulk of the work whilst the third party may offer specialist skills and through its independence could be used to verify DA obsolescence claims, since the DA may have a vested interest.

### 2.4 Strategy tools

A valuable tool to assist in the strategic decision making process is the decision triplet.
A significant discriminator in the OM strategy and decision making process is the use of the **cost**, **impact** and **probability** triplet. This basic aid to the decision making process should be employed during the analysis activity. The data supporting its application may, initially, be based solely on the judgment and experiences of the team but this should improve as the program develops.

In essence the triplet considers the following:

- the cost of the occurrence of an obsolescence event. This should include the total system life cycle cost addressing the design, production and implementation elements of the immediate system and any rework or support activity costs (cost of ownership).
- the impact of an event on the capability of the system. This should be considered with the customer or user and includes consideration of any reliability, maintainability and availability aspects.
- the probability of the event occurring. This includes possible environmental and safety legislation considerations. These three elements should be considered together to determine the classification of the obsolescence event. This classification may be used to determine the obsolescence strategy, the level of monitoring, whether further investigation is required and the periodicity of review.

### 3 Develop an obsolescence management plan

The first step in managing obsolescence is to assemble a comprehensive obsolescence management plan. This document should implement the strategy that will be applied during the life of the system or program. It should be initiated at the earliest phase of the program and reviewed at least at the end of each program phase for applicability. It may start as quite a small document but will develop and be refined as the lessons and decisions of the design and development phases are incorporated. As the program progresses it will develop into a through life record of the strategies employed, options considered and decisions made. It may incorporate the contractor's OM plan, which will be provided in response to the Invitation to Tender and contract.

The OM plan shall:

- achieve the optimum compromise between whole life costs, performance, availability and maintainability for the entire system
- cover all materiel, including hardware, software, tools, test equipment, human resources and training
- be compatible with the customer's current support arrangements
- provide a clear basis upon which obsolescence management requirements can be negotiated with suppliers and partners
- show consideration of the need for component or equipment requalification/re-certification following component or module substitution
- identify the linkages and interfaces into the risk and life cycle cost program activities

### 3.1 Scope and content of an obsolescence management plan

The generation of the plan and its contents are the responsibility of the project team and should reflect the scope, complexity and cost of the program. There are certain minimum requirements of the OM plan which are highlighted below.

It shall:

- identify obsolescence issues in the platform
- identify obsolescence standards to be used
- identify obsolescence performance metrics
- record the choice of strategy and provide detail of any plans and OM decisions
− for decisions made it shall include a note of the options considered, analyses and trade-off decisions behind any choice for later reference (subsidiary documentation should contain a full record of the factors)
− details of the stakeholder community personnel with authority for decision making on trade-offs between cost and impact. OM requires input from the customer, OEM, subcontractor and suppliers in some cases. A team with the correct stake holders needs to be assembled. The composition of this team will vary with the product and potentially the life cycle of the product. The team needs to establish a good working relationship so information and decisions can be quickly determined. The longer it takes to implement a solution reduces the chances it will be successful.
− Establishing a schedule and strategy to refresh the electronics will allow the capability upgrades as well as allow the product to be supported in the future. The strategy depends on the product and could be broken down into electrical/electronic parts, systems or sub systems and non-electrical/electronic parts systems or sub systems. This is also where the frequency of scanning the product breakdown should be set. The more often the product breakdown is scanned the less likely that an obsolescence alert will become a serious issue. However, the more often the scan is performed, the more expensive the process will become.
− Identification of a tool set to predict the life cycle of electronic components will enable a review a bill of materials and predict when an electronic component will no longer be supported. The tool should have the ability to read all levels of a product breakdown. The tool should have the ability to predict when a component or assembly may become obsolete.
− To effectively manage obsolescence issues there needs to be a source identified to receive notices from companies of discontinuances. Discontinuance notices alert customers that production is concluding for a specific part. The notices usually contain part numbers, last order and shipment dates, and minimum order quantities. Collection of discontinuance notices is essential to be proactive in the management of obsolescence.
− Establish a secure data repository for obsolescence data. Establishing a secure area where information can be exchanged and stored is essential. This protects the intellectual property of all parties. This data can be used again for similar products in the same stage of their life cycle.

4 Implementation of obsolescence solutions

When there is an obsolescence occurrence the choice of solution is wholly dependant on the program and market circumstances at that time. Therefore each occurrence should be treated as an individual event for which at least the following may need to be considered:
− reclamation
− last time buy
− life time buy
− re-manufacture
− use of an interchangeable item
− adaptive solutions
− reverse engineering
− new design

4.1 Budget for obsolescence management costs

It is essential that budgetary provision is made for the OM task itself. OM carries its own resource cost and this must be included in the program budget. In addition costs must be included for the resolution of the obsolescence issues that may arise. These will vary as the program develops and should be regularly reviewed. This cost forecasting effort should be linked to the program risk activity to ensure that a consolidated approach is taken and that items are not accounted for twice or not at all.

Applicable to: All

S3000L-A-15-00-0000-00A-040A-A

Chap 15

DMC-S3000L-A-15-00-0000-00A-040A-A_001_00_EN-US.doc

2010-04-01 Page 6
4.2 Linking obsolescence to the risk management activity
Managers must consider the operational risks over the life of the equipment associated with obsolescence:

− What would be the impact of equipment being unavailable due to lack of spares or of performance degradation due to substituted parts?
− What would be the likely cost of premature replacement or of other measures to circumvent obsolescence?
− What is the probability of obsolescence occurring (including considerations of technology advances and the introduction of new legislation)?

Obsolescence is a key contributor to any risk assessment. How the program will address obsolescence in its risk processes should be established early on in the program and reflected in the appropriate management plans. Having good visibility of the obsolescence risks is the first key step required to minimize the negative effects. This will allow the programmed management to identify and implement the most appropriate action to deal with any issue in the most cost-effective way. It is important to keep in mind that at the end it is always the system user the most affected during inservice phase in terms of readiness, supportability and life cycle costs. It should be noted that obsolescence is always a risk but it can create opportunities when alternative parts offer improved performance.

4.3 Obsolescence and life cycle costs
Obsolescence can lead to major changes in design, it may lead to changes in support philosophy, it may impact support arrangements. Because obsolescence can impact all phases of a program any review of solution options must consider the implications for the life of the program. Hence the need for OM to be linked to the LCC activities. A whole life approach must be considered when addressing cost issues. This should be reflected in the appropriate management plans.

4.4 Ensuring industry plans and manages obsolescence
Industry must plan for obsolescence from the very beginning of the design process including “obsolescence prevention” as another objective of the supportability engineering and logistics activities, maintaining this objective in all the phases of the program. Industry should:

− produce an OM plan
− conduct obsolescence critical analysis in the program to identify the system items with the highest risk of obsolescence and in accordance with a predefined criteria, using similar methodologies used for the safety hazard analysis and mission critical analysis
− adopts a pragmatic approach by putting into practice, as a continuous effort with all levels of industry participants, timely and cost-effective obsolescence engineering practices, management tools, methods and practices to avoid the negative effects of obsolescence in the program
− prepare obsolescence risk mitigation strategies for their incorporation in the Statement of Work (SOW) for all provisioning contracts during the system life cycle
− detail OM data requirements clearly within the statement of work and rights of access are addressed during commercial negotiations
− determines the obsolescence status of parts before its procurement
− All stakeholders plan the most cost effective time scales for in service technology updates/upgrades (technology road maps), managing obsolescence in existing equipment to extend their service life (this has to be in line to the logistics and also to the necessary qualification/certification processes)
− Apply practices already taken in other programs for obsolescence mitigation whenever it is deemed convenient and cost-effective

It is imperative that the Invitation to Tender for all phases of a program includes the requirement for an OM plan. The format and contents of the plan may be left to the discretion of the contractor but must as a minimum conform to the requirements of a recognized standard.
Intellectual property rights must be considered with regard to OM and in particular to ensure access to data required to support the OM strategy. The requirements of OM should also be considered in the context of an exit strategy, to ensure that all the data necessary to manage obsolescence is available in the event of a contract break.

One commercial strategy, which may be employed, is to pass responsibility for the management of obsolescence to the contractor completely. This hands-off solution can be adopted and in its simplest form consist of paying a fixed price to solve any future obsolescence arising. This is not at all an optimal technical solution but rather a more commercial or contractual approach, passing all liability for future obsolescence cases to industry during a specific period of time. In some cases, this could be accompanied by a redistribution of payment plan by putting forward some payments in order to cover potential obsolescence arising during a specific period of time. There are certain drawbacks to this approach. Although “risk is being transferred to the contractor” for a fee, care must be taken to ensure that there is at least limited visibility from the customer side so that he can see where action is being deferred or risk carried. These decisions could impact the customer on contract closure. Alternatively provision should be made to address obsolescence risk at the approach to contract conclusion, if this is achievable. In addition, during the fixed price negotiation, it may be difficult to justify or investigate costs unless the contractor proposal is supported by a detailed risk based obsolescence costed analysis.

5 Conclusion

There are many causes of obsolescence. The loss or impending loss of manufacturers or suppliers of critical items, raw materials, intellectual resources, exponential costs of scarce parts and materials, rapid changes in technology, uneconomical production requirements, competition, environmental or safety legislation all impact the sustainability of a product during it’s life cycle. OM is an activity intended to minimize the impact of this potential loss of supply through the identification, quantification and resolution of obsolescence and aims to achieve optimum cost-effectiveness. OM focuses on the system-wide application of risk management and is applicable to the entire program life cycle, becoming an integral part of the design, development, production and inservice support phases of the program.
Chapter 16

Inservice data feedback

Table of contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inservice data feedback</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>1</td>
</tr>
<tr>
<td>1  General</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Data sources</td>
<td>2</td>
</tr>
<tr>
<td>2  Data usage</td>
<td>3</td>
</tr>
<tr>
<td>3  Conclusion</td>
<td>3</td>
</tr>
</tbody>
</table>

List of tables

1 References ................................................................................................................. 1

List of figures

1 Acquisition logistics main business processes ......................................................... 2

References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1000D</td>
<td>International specification for technical publications using a common source database</td>
</tr>
<tr>
<td>S2000M</td>
<td>International specification for material management</td>
</tr>
<tr>
<td>S4000M</td>
<td>International procedure handbook for the development of scheduled maintenance programs for military aircrafts</td>
</tr>
<tr>
<td>S5000F</td>
<td>International specification for operational and maintenance data feedback</td>
</tr>
</tbody>
</table>

1 General

1.1 Introduction

The feedback of inservice data is one of the most important functions of inservice support. It enables fleet and support managers and defense system manufacturers to perform a thorough analysis of operational and maintenance performance of a product.

The results of this analysis can be the basis for

− enhancement of the operational effectiveness
− improvement of the product by modifications and retrofit activities
1.2 Objective

This chapter should be used as guidance to establish the requirements for and the importance of the inservice operational maintenance data.

The feedback of inservice data is an essential element in any Integrated Logistics Support (ILS) program. The data feed from inservice operations and maintenance has many uses. The data forms the basis for the empirical validation, update and correction of the theoretical, calculated and predicted values that have been established within the Logistic Support Analysis (LSA) activities. The values measured through the inservice operation of the product can be used to verify and update the values in the LSA to ensure that the LSA stays current with the actual operation of the product.

1.3 Data sources

The following are examples of inservice data that can be used to update existing Logistic Support Analysis programs. This can be through updates to the Failure, Modes Effect and Criticality Analysis (FMECA), adjustments to the maintenance schedules, update and revision to maintenance tasks:

- Location of repair and operating environment
- Operational environment
- Operational regime
- Maintenance level
- Skill level/Shop that performed the repair
- Maintenance man hours and elapsed time
- Failure/Damage cause
- Induced failures/damage
- Operational condition or type of mission when the failure occurred
- Repair actions taken
- Materials consumed
- Replacement parts
- Field service reports (maintainer/customer feedback, support equipment performance/accuracy, etc)
- Operator reports/debrief
- Equipment availability incident
- Troubleshooting data (includes unresolved failure detection – cannot duplicate)
- Failure rates
- No defect rates
- Item identifier (part number, serial number, configuration, version)
- Item configuration (service bulletin, in service modifications, retrofit, etc.)
- When discovered (used to verify and optimize maintenance intervals)
- Sampling study (in service observation/tracking)
- Accumulated flight and/or operational hours
- Reason for action
- Fault symptoms or indication of failure
- Inventory/Location of material/parts/hardware
- Logistics downtime – administrative delays
- Logistic delays
- Supplier/Provider feedback
- Data cleansing/validation of in service data
- Warranty information
- Customer local directives/work around/interim procedures
- Technical publication change requests

2 Data usage

Analysis and compilation of In Service Feedback data may provide a means to:

- Accurately define/forecast spares provisioning at all maintenance levels
- Justify the need for scheduled/preventative maintenance task. The cycle of an existing task can be updated from in service data using a sampling program.
- Update manpower requirements & staffing levels
- Correct the required quantities of support resources
- Update other maintenance planning related documents
- Identify items that may require product improvement

3 Conclusion

The feedback of inservice data is not only for logistic analysis purposes. The data that is captured inservice is also used for fleet planning and management, product liability and warranty claims, material and stock forecasting and provisioning.

The overall aim to be achieved through inservice data feedback is to increase of availability of the product and optimize its effectiveness and to establish better information and data for future contract and programs.
## Chapter 17

**Disposal**

### Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposal</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1. General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Purpose</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>2. Disposal tasks during Product life cycle phases</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Concept definition phase</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Development phase</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Inservice phase</td>
<td>5</td>
</tr>
<tr>
<td>2.4 Disposal phase</td>
<td>5</td>
</tr>
<tr>
<td>3. Disposal analysis</td>
<td>5</td>
</tr>
<tr>
<td>3.1 Disposal analysis activities</td>
<td>6</td>
</tr>
<tr>
<td>3.1.1 Define a disposal strategy</td>
<td>6</td>
</tr>
<tr>
<td>3.1.2 Integration of disposal requirements in design</td>
<td>6</td>
</tr>
<tr>
<td>3.1.3 Disposal method analysis</td>
<td>7</td>
</tr>
<tr>
<td>3.1.4 Hazards</td>
<td>8</td>
</tr>
<tr>
<td>3.1.5 Configuration management</td>
<td>8</td>
</tr>
<tr>
<td>3.1.6 Monitor and control of sub-suppliers</td>
<td>8</td>
</tr>
<tr>
<td>3.1.7 Task input</td>
<td>8</td>
</tr>
<tr>
<td>3.1.8 Task output</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Disposal operation task identification</td>
<td>9</td>
</tr>
<tr>
<td>3.2.1 Disposal process breakdown analysis</td>
<td>9</td>
</tr>
<tr>
<td>3.2.2 Task identification</td>
<td>10</td>
</tr>
<tr>
<td>3.2.3 Disposal hazard analysis</td>
<td>10</td>
</tr>
<tr>
<td>3.2.4 Task input</td>
<td>11</td>
</tr>
<tr>
<td>3.2.5 Task output</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Level of disposal analysis</td>
<td>11</td>
</tr>
<tr>
<td>3.4 Disposal task analysis</td>
<td>11</td>
</tr>
<tr>
<td>3.4.1 Task description</td>
<td>11</td>
</tr>
<tr>
<td>3.4.2 Task input</td>
<td>12</td>
</tr>
<tr>
<td>3.4.3 Task output</td>
<td>12</td>
</tr>
<tr>
<td>4. Product disposal file</td>
<td>13</td>
</tr>
<tr>
<td>4.1 PDF content</td>
<td>13</td>
</tr>
<tr>
<td>4.2 PDF compilation process</td>
<td>13</td>
</tr>
<tr>
<td>4.3 PDF data elements</td>
<td>15</td>
</tr>
<tr>
<td>5. List of regulations</td>
<td>18</td>
</tr>
<tr>
<td>5.1 European Union directives</td>
<td>18</td>
</tr>
<tr>
<td>5.2 Transportation regulations</td>
<td>18</td>
</tr>
<tr>
<td>5.3 Support equipment and ground equipment</td>
<td>18</td>
</tr>
<tr>
<td>5.4 Navy equipment</td>
<td>19</td>
</tr>
<tr>
<td>5.5 Air equipment</td>
<td>19</td>
</tr>
<tr>
<td>5.6 Ammunition</td>
<td>19</td>
</tr>
<tr>
<td>5.7 Nuclear waste</td>
<td>20</td>
</tr>
</tbody>
</table>
1 General

1.1 Introduction

Disposal is both a business process and the name of a phase in a product life cycle. This chapter describes the disposal business process.

Regarding disposal, it is no longer acceptable to dump worn out equipment into the sea or anywhere else that might allow it to cause harm to human health or the environment. Disposal needs to be performed in a controlled manner that does not impact human health and does not cause the contamination of land, water or air.

It must be stated that:

- disposal is an event that occurs only once during a product life cycle. Once disposed, the product no longer exists
- the probability that disposal will actually occur is 100%, at some point in some manner

This leads to the conclusion that disposal requirements must be implemented into the design from the very beginning of a development program. The general disposal principles to be applied are:

- to have as much material as possible to be recycled
- to minimize landfill waste resulting from the product’s lifecycle

Therefore, disposal must be considered as one of the major business processes to be implemented within the framework of an ILS/LSA program development strategy.

To dispose of a product means to phase it out at the end of the product's service life. This paragraph deals with disposal of products. Typically the disposal of products happens during the last phase of their life cycle, however, disposal must be considered from the very early phases of every program that acquires a product.
1.2 Purpose

This chapter provides guidance for developing a disposal procedure applicable for a product (existing or under development), that can be demonstrated to be safe, efficient, environmentally acceptable and cost effective.

It also provides guidance to analyze required operations and disposal tasks for the equipment to:

- Identify logistic support resource requirements for each task
- Identify new or critical logistic support resource requirements
- Identify transportability requirements
- Identify support requirements that exceed established goals, thresholds, or constraints
- Provide data to support participation in the development of design alternatives to reduce disposal costs and optimize logistic support resource requirements
- Provide guidelines on what must be done during each product phase and how to properly measure the effort
- Provide source data for the preparation of required ILS documents (eg disposal instructions, training program, manpower and personnel lists)

1.3 Scope

Due to growing concerns, disposal activity must consider the following issues:

- Destruction/neutralization of toxic substances such as carcinogenic, mutagenous, reprotoxical that harm humans or the environment for short or long periods of time (eg chemicals, radioactive substances)
- Allow a sustainable development by recycling materials (eg reuse of assemblies, raw material) or converting them into energy (eg burning with toxic gas processing)
- Demilitarization of defense products to avoid weapons proliferation and use by terrorist groups

This applies to:

- the product disposal at end of the service life
- the waste generated throughout the product operation life cycle including, but not limited to:
  - Ozone depleting substances (CFC, HCFC, halons)
  - Global warming effect (sustainable development)
  - Health care waste and sanitary waste
  - Stores (eg paints, solvents)
  - Ballast water and sediments
  - Anti-fouling paints (eg biocides)
  - Oily bilge water (sludge)
  - Wastewater (black and grey water)
  - Galley food waste
  - Solid waste (eg glass, paper, plastics)
  - Exhaust emissions (eg NOx, SOx, COx)
  - Spares and consumables (eg batteries, nuclear sources, oils)
- All of the resources deployed and used to operate and support the mission product (eg infrastructures, facilities, support equipment)

The disposal process impacts the product's following phases:

- Operation and Support phase
  - Take the necessary provisions to design safe Operation and Support tasks against dangerous substances contained in the product,
  - Disposal process of consumables and items removed as a result of performance of Operation and Support tasks (health and environment impact limitation).
This information shall be available in the LSA database and in the product’s technical publications (Operation and Support tasks).

- Disposal phase
  - Dismantling a disassembling
  - Demilitarization
  - Dispose of products (eg energetic recovery, material recovery, re-use, landfill waste)

This information shall be available in Product Breakdown Structure (PBS) and Product Disposal File (PDF).

2 Disposal tasks during Product life cycle phases

2.1 Concept definition phase
A product disposal strategy is created along with a concept of operation and a support policy.

This strategy must make provisions to appropriately address the following:

- Sustainable design to limit the use of scarce resources, and the environmental impact of product disposal
- Identification of health and environmental regulations that are applicable (refer to Para 5)
- Identification of the list of black and grey substances that will be respectively forbidden and limited in use and quantity
- Definition of the method for tracking the dangerous substances, applicable in house and flowed down to suppliers
- Estimate of the Rough Order of Magnitude of product disposal cost

2.2 Development phase
The disposal activity during development phase is twofold:

- Design the disposal
  This includes all plans made during the product design in a sustainable development approach, such as limiting the use of scarce resources, reducing the environmental impact of Product disposal (eg no toxic substances, identification of materials for easy recycling).

  For this, a PDF is compiled (refer to Para 4).

  Cancerogenous, Mutagenous, Reprotoxical substances used in the product design are recorded in the Product breakdown as part of Product Data and Configuration Management (CM). Chemical substances are identified by their Chemical Abstract Substance code (or equivalent), according to the REACh European regulation requirements.

  As the product design matures, the PDF information is updated.

- Dispose of the design
  This includes all plans made during development to further prepare for product disposal phase. It includes the following:
  - Perform a product dismantling task analysis, where tasks required to dispose of the Product are identified and characterized (eg tasks duration, required resources).
    This work can be considered as an extension of the Maintenance Task Analysis (MTA) performed as part of the product Logistic Support Analysis (LSA), using the same structured approach.
  - Assess the impact of disposal on product logistic support elements such as additional support equipment required to perform the product disposal, describe the disposal tasks in the technical publications, training courses to teach operators how to dispose of the product (including health and safety cautions).
• Estimate the cost of the product disposal. This work can be a part of the product global Life Cycle Cost (LCC) analysis work, using the same structured approach.

2.3 Inservice phase
During this phase:
  – the PDF is updated to take into account the following:
    • Design changes in the product during its phase of Operation and Support
    • Evolution of applicable regulations (e.g., new banned substances, strengthening of health and safety)
  – resources required for product disposal are produced and delivered
  – disposal demonstration can be performed on product prototypes to validate:
    • Relevancy and completeness of information compiled in the PDF.
    • Dismantling and disposal operations (e.g., people skills, technical publications contents, support equipment adequacy)

Equipment to be replaced (e.g., batteries, oils) according to the maintenance plan must be disposed of in compliance to disposal requirements identified in the disposal analysis performed during the previous product life cycle phases.

2.4 Disposal phase
During this phase:
  – The disposal strategy defined during the concept definition phase is implemented to dispose of the product (e.g., dismantling, recycling, energetic valorization)
  – The disposal operations are facilitated due to:
    • A product design that is disposal oriented (design the disposal)
    • An available set of resources developed in order to perform the product disposal (dispose of the design)
  – Lessons learned are collected (e.g., cost, best practice, things to avoid) to be fed back to other programs

3 Disposal analysis
As shown in the figure below, the LSA activities logic to determine the disposal process, disposal tasks, and support resources required, starts with the disposal analysis that needs to be tailored to the program.

This analysis applies to both:
  – Product subassemblies to be disposed of all along the product life cycle, as a result of the product maintenance plan implementation
  – The entire product, at the end of its operational life
3.1 Disposal analysis activities

This paragraph describes the LSA activities to be considered and tailored for the disposal at the end of a product’s life. The task logic of the disposal analysis can be illustrated as given in Fig 2:

3.1.1 Define a disposal strategy
Define the disposal criteria for each item in the PBS and establish a Design Solution.

3.1.2 Integration of disposal requirements in design
Upon the development of a new product, the disposal requirements must be balanced against those for performance, reliability, safety and life cycle cost.

The design principles shall be a guide for design work and design considerations. Materials used in the design of the product and its packaging shall be selected to meet planned service life requirements.
3.1.2.1 Recommended materials

Materials to be used in the design of the product and its packaging shall be selected to minimize:

- hazards to personnel or property during preparation for disposal
- hazards to personnel, property or the environment as a result of the processes used in disposal
- residual hazards from the products of the disposal process

All materials shall be fully specified, along with their associated hazards, in their primary state and during the stages of the disposal process. Material identification and marking processes must be defined. This information should be recorded in the PDF (refer to Para 4).

3.1.2.2 Accessibility and extraction of components and hazardous materials

Provide design recommendations with requirements for a minimum of non-standard processes, tools and technical expertise for access and extraction, and identify any non-standard processes or tools needed for access to components and hazardous materials, as well as any hazards associated with such processes.

3.1.2.3 Select components and materials to facilitate safe recovery and recycling

Identify the components and the base materials likely to be recovered from the product and their potential for recycling. The assessment shall consider the possibility of degradation and contamination of components and materials during their service life such as to render recovery and recycling no longer viable.

3.1.2.4 Complete material list

List all materials used in the product in a supplier material list and make it available to all personnel it may concern in the program. As a minimum, the list shall include, but not be limited to:

- Components
- Materials
- Compatibility aspects
- Residual products used by the product
- Residual products at planned disposal
- Impact on the environment at planned disposal
- Applicable health and environmental regulations

3.1.3 Disposal method analysis

1. Identify potential alternative applications (re-use) for the product, sub-products, assemblies and components, including software. Demonstrate that they are technically achievable, environmentally acceptable and economically viable

2. Identify components and material that must be recycled

3. Identify components and material that must be recovered

4. Identify components and material that will be land waste

5. For each alternative above, define the residual products. Refer to Table 2.

6. Document the results in a disposal PBS

7. Disposal of sensitive information

When products to be disposed of contain proprietary, confidential or classified software and data, the dismantling procedure shall include the two following steps:

7.1 The transfer of the necessary historical information to a media or other storage device for proper preservation or historical archiving,
7.2 The destruction of the data/software media or the erasure of the sensitive, proprietary, and classified information and software in an adequate manner so as to ensure that such information/software cannot be recovered by third parties, in accordance with the proper security regulations.

<table>
<thead>
<tr>
<th>Residual products</th>
<th>Re-useable/recyclable [g]</th>
<th>Combustible [g]</th>
<th>Waste [g]</th>
<th>Accumulated weight [g]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>20</td>
<td>2000</td>
<td>3000</td>
<td>5020</td>
</tr>
</tbody>
</table>

3.1.4 **Hazards**

Identify all hazards associated with the design and in particular:

- Provide an estimate of the likely levels of noise, flash, and toxic or corrosive fumes produced upon operation/firing/launching.
- Identify any disposal activity hazards associated with the product design that requires additional safety arrangements other that the existing disposal resources.
- Identify any residual hazards from the disposal of the product that requires additional safety arrangement other than the existing disposal resources.
- Consider possible means of fail safe.
- Identify any features that might pose an unusual hazard to Explosive Ordnance Disposal operatives, for example, anti-tampering/interference fuses or energetic materials with particularly high levels of sensitivity. This assessment must consider the effects of aging on the product.

3.1.5 **Configuration management**

For any component/design change, assess its impact on the:

- product life extension
- reduction of health and environment hazards

and demonstrate that such changes are both technically achievable and economically viable.

3.1.6 **Monitor and control of sub-suppliers**

Product elements obtained from suppliers shall meet the disposal requirements. This effort applies to equipment obtained from any supplier, regardless of tier. Review all sub-supplier contracts and perform evaluations of their disposal procedures.

If Government Furnished Equipment is used, they shall be treated as ordinary sub-supplier items. The same requirements shall be applied as for those suppliers.

During the subsequent phases of the development program, sub-supplier specifications shall incorporate aspects such as:

- Disposal requirements
- Design constraints and requirements
- Material Safety Data Sheets (MSDS)

3.1.7 **Task Input**

- Disposal strategy
- Disposal requirements
- Disposal criteria
- PBS
- MSDS detailing:

Applicable to: All
• The product name used on the label, and chemical and common names of ingredients which have been determined to be health hazards, and which comprise 1% or greater of the composition, except carcinogens shall be listed if the concentrations are 0.1% or greater.
• The chemical and common names of all ingredients which have been determined to present a physical hazard when present in the mixture.
• Relevant physical and chemical characteristics of the hazardous chemical (such as vapor pressure, flash point).
• Relevant physical hazards, including the potential for fire, explosion, and reactivity.
• Relevant health hazards, including signs and symptoms of exposure, and any medical conditions generally recognized as being aggravated by exposure to the chemical.
• The primary routes of entry into the body.
• The Occupational Safety and Health Administration permissible exposure limit and American Conference of Industrial Hygienists threshold limit value. Additional applicable exposure limits may be listed.
• Whether the hazardous chemical is listed in the National Toxicology Program Annual Report on Carcinogens (latest edition) or has been found to be a potential carcinogen in the International Agency for Research on Cancer Monographs (latest editions), or by Occupational Safety and Health Administration.
• Precautions for safe handling and use, including appropriate hygienic practices, protective measures during repair and maintenance of contaminated equipment, and procedures for clean-up of spills and leaks.
• Appropriate control measures, such as engineering controls, work practices, or personal protective equipment.
• Emergency and first aid procedures.
• The date of preparation of the MSDS or the last change to it.
• The name, address and telephone number of the chemical manufacturer, importer, employer or other responsible party preparing or distributing the MSDS, who can provide additional information on the hazardous chemical and appropriate emergency procedures, if necessary.

3.1.8 Task output
- Scope of the disposal business process to be applied.
- A disposal PBS list, detailing which items should go for re-use, recycling, recovery or landfill waste.

3.2 Disposal operation task identification
3.2.1 Disposal process breakdown analysis
Define a representative set of activity sequences to identify all required services that correspond to anticipated disposal process of a product and environments.

The task shall list the method recommended for disposal of the product items. Disposal involves two aspects:
- Dismantling the product
- Disposal of materials and contaminated items, including contaminated solvents

A breakdown of the recommended process shall be updated as details of the design become available.

This requires the use of the following data elements:
- a disposal task code
- disposal recovery codes for the Source Maintenance & Recoverability (SMR) code, coding for the method by which the product is intended to be disposed of (eg recycled, energy recovered, landfill waste)
3.2.2 **Task identification**

Identify and document the operations/tasks that must be performed for disposal of the equipment. These tasks shall be identified to a level commensurate with design, including disassembly diagrams, step-by-step procedures, safety precautions and component and material tables, taking into account possible changes due to aging, and final destination of all products. Any special tools and equipment required and any limitations on locations are to be identified.

A task inventory shall be prepared for the equipment. This task inventory shall identify all tasks that operators, maintainers, or support personnel must perform with regard to disposal of the equipment (hardware and software) based on the disposal process breakdown analysis, and scenarios/conditions. Tasks shall be identified to a level of detail commensurate with design and the disposal scenario development. The task inventory shall be organized in terms of a task classification which defines scenario/conditions, function, job and duty, task and subtask and task elements. The task inventory shall be composed of task descriptions, each of which consists of:

- An action verb which identifies what is to be accomplished in the task
- An object which identifies what is to be acted upon in the task

Task descriptions shall be clear and concise. Hazardous materials, generation of waste, release of air and water pollutants and environmental impacts associated with each task shall be identified. Where the same task appears in the duty of more than one job, and is therefore identified in a collective task for training purposes, it will be identified as such within the task inventory. All verbs shall be unambiguously defined within the task classification.

3.2.3 **Disposal hazard analysis**

The hazard analysis of the chosen disposal process shall identify the hazards related to the process and its products, and shall quantify the associated risks. The results of the analysis can be presented using a matrix, where the risk scores are:

- Serious = 2
- Moderate = 1
- Harmless = 0

<table>
<thead>
<tr>
<th>Risks</th>
<th>Type of Risk</th>
<th>Weight</th>
<th>Hazards from planned disposal process</th>
<th>Total (Risk x Weight)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Safety</td>
<td>3</td>
<td>Detonation of warhead at dismantling of missile</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>Health</td>
<td>3</td>
<td>Toxic gases produced at burning of launch rocket motor propellant</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>Environmental</td>
<td>2</td>
<td>Environmentally polluting gases produced at burning of launch rocket motor propellant</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3 Risk score example

The hazard analyses of the product shall include a description of the product with its configuration. The composition and mass of each material shall be listed along with an aggregate mass of all materials in the item. The hazards associated with each material shall be identified along with any special handling requirements as follows.

Both primary hazards and secondary hazards (those derived from material changes that occur as a result of the processes used in disposal) shall be described in the analyses. All materials to be transported shall be accompanied by safety data sheets.
Risks to be listed and considered are, for example:

- **Explosive hazards**
  Risks associated with explosive materials in the product

- **Fire hazards**
  All flammable and oxidizing materials

- **Caustic hazards**
  All caustic materials

- **Ingestion, absorption, inhalation and adsorption**
  All materials that have a physiological effect on personnel or may damage property. The list shall indicate the physical form of the material, the means of attack and the protection required

- **Trauma inducing**
  All non-explosive energy stores such as batteries, electrical and electromagnetic generating devices, capacitors, sources of static charge, springs under tension or compression, and any material that can transfer energy upon release, sufficient to cause trauma. The list shall indicate likely energy output, likely effect and the protection required

- **Environmental hazard**
  Any potential environmental hazard not already covered. These shall include potential damage to the atmosphere, soil and groundwater

### 3.2.4 Task input

- Delivery identification of any data item required.
- Identification of equipment hardware, software and firmware on which this task will be performed and the indenture levels to which this analysis will be carried.
- Identification of the levels for disposal of hardware, software and firmware which will be analyzed during performance of this task to identify operations and tasks.
- Any documentation requirements over and above the LSA database, such as functional flow diagrams or drawings.
- Description of equipment concepts under consideration.
- Disposal method (disposal PBS).

### 3.2.5 Task output

- A task inventory documented in the LSA database, or equivalent format approved by the project, identifying disposal operational tasks requirements, to include task descriptions on Product hardware software and firmware and to the indenture levels specified by the project.
- Identification of any hazard risks involved in satisfying the disposal operational tasks requirements.

### 3.3 Level of disposal analysis

The level of disposal will be driven by the hazardous substances location in the product breakdown. The product must be dismantled and disassembled to a level of subassemblies which can be disposed of using the same disposal method as defined in Para 4.

### 3.4 Disposal task analysis

#### 3.4.1 Task description

- Conduct a detailed analysis of each operation, maintenance and support task contained in the task inventory and determine the following:
• Logistic support resources required (considering all ILS elements) to perform the task.
• Elapsed time and man-hours of the equipment intended disposal facility/environment.
• Maintenance level assignment based on the established support plan.
• Environmental impact of the tasks, including use of hazardous materials, generation of hazardous waste and its disposal, and the release of air and water pollutants.

− Identify new or critical logistic support resources required to perform each task, and the hazardous materials, hazardous waste and environmental impact requirements associated with those resources. New resources are those which require development in order to disassemble the new equipment. These can include support and test equipment, facilities, new or special transportation products, new computer resources, and new test or inspection techniques. Critical resources are those which are not new but require special management attention due to schedule constraints, cost implications, or known scarcities. Unless otherwise required, document new and modified logistic support resources in the LSA database, or with equivalent documentation approved by the project, in order to provide a description and justification for the resource requirement.
− Conduct a transportability analysis on the equipment and any sections thereof when sectionalization is required for transport.
− Document the results in the LSA database, or by an equivalent format approved by the project.

3.4.2 Task input
− Identification of equipment hardware and software on which this analysis will be performed
− Identification of indenture levels to which this analysis will be performed
− Identification of the levels of disposal operations that will be documented during performance of this task
− Known or projected logistic support resource shortages
− Any supplemental documentation requirements over and above the LSA database (eg transportability clearance diagrams)
− Delivery identification of any data item required
− Information available from the project relative to:
  • Existing and planned personnel skills, capabilities, and program of instruction
  • Lists of standard support and test equipment
  • Facilities available
  • Training devices available
  • Existing transportation products and capabilities
− Description of personnel capabilities (target audience) intended for disposal of the equipment at each level of disposal
− Operations and disposal task requirements from task (disposal operations task identification)
− Recommended disposal plan for the equipment from task (level of disposal analysis)
− Supportability and supportability related design goals and requirements

3.4.3 Task output
− Completed LSA database on equipment hardware and software to the indenture level specified by the project, or equivalent format approved by the project.
− Identification of new or critical logistic support resources required to operate and maintain the new product.
− Alternative design approaches where tasks fail to meet established goals and constraints for the new equipment or where the opportunity exists to reduce disposal costs or optimize logistic support resource requirements.
− Identification of management actions to minimize the risks associated with each new or critical logistic support resource requirement.
− Validation of key information documented in the LSA database.
Output summaries and reports as specified by the project containing all pertinent data contained in the LSA database at the time of preparation.

An LSA database that is updated as better information becomes available and as applicable input data from other product engineering programs is updated.

4 Product disposal file

The Product Disposal File (PDF) is intended to compile all data information regarding product disposal. It must be updated throughout the product lifecycle in order to reflect product changes (configuration management), maintenance plan evolutions, and health and environmental regulations modifications.

This file must allow the product program management to clarify the operational limits of the authorized substances over a given period, in order to anticipate the research of alternative solutions which are more suitable in terms of:

- Sustainable development
- Health and safety impact on humans and environment

4.1 PDF content

Whatever the product, the PDF may compile information such as, but not limited to the following:

- Product breakdown and associated functional and physical description.
- Related to the product breakdown (with illustrations wherever possible), identification of components, products or ingredients (eg containing hydrocarbon, lubricating, paintings).
- For each dangerous substance identified in the product, list of the risks for people and environment.
- For each type of product subassembly, according to the type of technology (eg electronic, mechanic, hydraulic and pyrotechnic) and the contained substances, identify their possible processing at disposal.
  - those immediately reusable on another product
  - those containing high value material (eg gems, precious metals)
- The methods of elimination that will make the difference between potentially harmful substances and harmless material.
- Conditions for disassembly and the destination of the components (eg re-use, recycling, energetic valorization).
- The procedures of disassembly down to the required level of depth, such that all the materials (including in particular those harmful or dangerous) can benefit from the processing and their mode of elimination.
- The rules, instructions, measurements or provisions to apply that guarantee safe practices at a sufficient level.
- The collection of all safety data for all the harmful or dangerous materials composing the entire product.
- The procedures for dismantling and disposal, detailed step by step.
- A list of proposed organizations certified for proper recycling and elimination of the dangerous residues, according to the applicable regulation.

4.2 PDF compilation process

The PDF compilation process is shown in the flowchart given in Fig 3:
Type of Product
  Initialize the PDF compilation
    Disposal data elements initialization
      Product design status
        No
          PDF already exists?
            Yes
              - Disposal strategy
              - Product description
              - Dismantling process
            No
              No
                Dangerous substances identification and characterisation
          No
            Disposal data elements already exists?
              Yes
                Product design and configuration file
                  No
                    Product design change?
                      Yes
                        - Disposal strategy
                        - Product description
                        - Dismantling process
                        - Disposal information
                        - Dangerous substances identification and characterisation
                        - Product design and configuration
                      No
                        Product Disposal File
                        ICN-B6865-S3000L0091-001-01

Fig 3 PDF compilation process
4.3 PDF data elements

This paragraph presents a proposed list of data elements to be compiled in the PDF. This list is not limitative, and must be agreed to on a case by case basis, depending upon the context of each specific program and contract.

Key of PDF Columns Header

Text in *italic* refers to data elements from the S3000L standard database

<table>
<thead>
<tr>
<th>Header</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsystem</td>
<td>Identifies the subunit to which the item belongs in the Product</td>
</tr>
<tr>
<td>Assembly identifier</td>
<td><em>Part_identifier</em></td>
</tr>
<tr>
<td>Item</td>
<td>Name of component according to drawing</td>
</tr>
<tr>
<td>Part name</td>
<td><em>Part_name</em></td>
</tr>
<tr>
<td>Article number</td>
<td>Refers to drawing No. for unequivocal identification or (for some components such as screws and adhesives) other article number</td>
</tr>
<tr>
<td>Part number</td>
<td><em>Part_identifier</em></td>
</tr>
<tr>
<td>Material Denomination</td>
<td>Short description of material in component (if known)</td>
</tr>
<tr>
<td>Material description</td>
<td><em>Part_material_description (ADD Material Class TO THE MODEL)</em></td>
</tr>
<tr>
<td>Cat</td>
<td>The main substances category: black (forbidden), grey (authorized with limitation), green (authorised with no limitation)</td>
</tr>
<tr>
<td>Material Category</td>
<td><em>(Include date when recorded)</em></td>
</tr>
<tr>
<td>CAS</td>
<td>Chemical Abstract Substance identification number</td>
</tr>
<tr>
<td>Material Code</td>
<td><em>(Material hazardous classification)</em></td>
</tr>
<tr>
<td>Phrase of Risk</td>
<td>Phrases of risk (as per REACh regulation safety Data Sheet data)</td>
</tr>
<tr>
<td>Type of Risk</td>
<td><em>Risk_description associated with the Material class</em></td>
</tr>
<tr>
<td>MSDS ref.</td>
<td>Reference of the associated Material Safety Data Sheet</td>
</tr>
<tr>
<td>MSDS</td>
<td><em>Document classified as ‘Material Safety Data Sheet, assigned to the Material Class in the role of ‘Reference’</em></td>
</tr>
<tr>
<td>Service life</td>
<td>Product service life</td>
</tr>
<tr>
<td>Service life</td>
<td><em>Minimum_product_service_life associated with Part</em></td>
</tr>
<tr>
<td>Mass (g)</td>
<td>Weight of component (in grams)</td>
</tr>
<tr>
<td>Mass</td>
<td><em>Quantity of material included in Part</em></td>
</tr>
<tr>
<td>Reg.</td>
<td>Applicable environmental regulations</td>
</tr>
<tr>
<td>Regulations</td>
<td><em>Document assigned in the role of ‘Regulation’</em></td>
</tr>
<tr>
<td>SLA No.</td>
<td>Serial number of component</td>
</tr>
<tr>
<td>Serial Number</td>
<td><em>Material included in a specific serial number range for the Part</em></td>
</tr>
<tr>
<td>Risk factor (SLA)</td>
<td>Judged risk factor (criticality) of component (1 to 4 scale)</td>
</tr>
<tr>
<td>Risk Criticality</td>
<td><em>Risk Criticality</em></td>
</tr>
<tr>
<td>Critical property</td>
<td>The most important property for the function of the component</td>
</tr>
<tr>
<td>Justification</td>
<td><em>Material_justification_description associated with Material usage Part</em></td>
</tr>
<tr>
<td>Proceedings in connection with use</td>
<td>Describes how the component is to be disposed of during the inservice phase</td>
</tr>
<tr>
<td>Disposal concept in use</td>
<td><em>Derived from any of eg Data_module or Task_overall_description or Task_requirement_specification (related to the Part)</em></td>
</tr>
</tbody>
</table>
### Header

<table>
<thead>
<tr>
<th>Header</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposal concept end of life</td>
<td>Derived from any of eg Data_module or Task_overall_description or Task_requirement_specification (related to the Part)</td>
</tr>
<tr>
<td>Waste products in use</td>
<td>States the waste products when the component is disposed of according to the procedure during use or after being used Narrative related to the Part</td>
</tr>
<tr>
<td>Waste products disposal</td>
<td>States the waste products when the component is disposed of according to the procedure for planned disposal. Narrative related to the Part</td>
</tr>
<tr>
<td>Environmental aspects, use</td>
<td>Classification of environment aspects during use. Refer to codes in the &quot;Environment aspect codes&quot; table below Classification related to Part</td>
</tr>
<tr>
<td>Environmental aspects, planned disposal</td>
<td>Classification of environment aspects at end of life. Refer to codes in the &quot;Environment aspect codes&quot; table below Classification related to Part</td>
</tr>
<tr>
<td>Task</td>
<td>Dismantling tasks characterization and description, identifying required resources and associated safety cautions to be taken to preserve human beings and environment. Task_requirement_specification</td>
</tr>
</tbody>
</table>

### Table 5 Environment aspect codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>tox</td>
<td>Harmful to environment according to note g and h in MBT04072. (toxic, bioaccumulation and/or difficult to decompose)</td>
</tr>
<tr>
<td>acid</td>
<td>Acidification</td>
</tr>
<tr>
<td>ozon</td>
<td>Dangerous for ozone layer</td>
</tr>
<tr>
<td>gwp</td>
<td>Greenhouse effect</td>
</tr>
<tr>
<td>mw</td>
<td>Material waste</td>
</tr>
<tr>
<td>pos</td>
<td>Environmental aspects, positive</td>
</tr>
<tr>
<td>ene</td>
<td>Energy regaining by burning</td>
</tr>
<tr>
<td>mr</td>
<td>Material recycling</td>
</tr>
</tbody>
</table>

Applicable to: All

S3000L-A-17-00-0000-00A-040A-A

Chap 17

DMC-S3000L-A-17-00-0000-00A-040A-A_001_00_EN-US.doc

2010-04-01 Page 16
<table>
<thead>
<tr>
<th>Item</th>
<th>Article number</th>
<th>Material denomination (specification)</th>
<th>Service life</th>
<th>Mass (g)</th>
<th>SLA No.</th>
<th>Critical Property (important for service life qualification)</th>
<th>Proceedings in connection with</th>
<th>Waste products in connection with</th>
<th>Environmental aspects</th>
<th>Task</th>
<th>Last updated SLA/Environ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outside tube</td>
<td>411349</td>
<td>PC 30% glass fibre Black</td>
<td></td>
<td>12.8</td>
<td>4</td>
<td>Mechanical strength</td>
<td>Thrown Burned/ recycling</td>
<td>Stable</td>
<td>COx, PC</td>
<td>mw</td>
<td>ene, gw, phtr</td>
</tr>
<tr>
<td>Plug</td>
<td>5229068</td>
<td>PC 30% glass fibre Black</td>
<td></td>
<td>4.1</td>
<td>4</td>
<td>Mechanical strength</td>
<td>Thrown Burned/ recycling</td>
<td>Stable</td>
<td>COx, PC</td>
<td>mw</td>
<td>ene, gw, phtr</td>
</tr>
<tr>
<td>Support Stand</td>
<td>411349</td>
<td>PC 10% glass fibre Black</td>
<td></td>
<td>26.7</td>
<td>4</td>
<td>Mechanical strength</td>
<td>Thrown Burned/ recycling</td>
<td>Stable</td>
<td>COx, PC</td>
<td>mw</td>
<td>ene, gw, phtr</td>
</tr>
<tr>
<td>Slotted screw MCS M4X25</td>
<td>6436317</td>
<td>Stainless Steel IS0 1207</td>
<td></td>
<td>3.0</td>
<td>10</td>
<td>Mechanical strength</td>
<td>Thrown Recycling</td>
<td>Steel</td>
<td>Steel</td>
<td>mw</td>
<td>mr</td>
</tr>
<tr>
<td>Connector cover</td>
<td>2799541</td>
<td>Aluminium EN 1706 AC42000076 Chromated Cr</td>
<td></td>
<td>5.0</td>
<td>306</td>
<td>Mechanical strength</td>
<td>Thrown Recycling</td>
<td>Al</td>
<td>Al</td>
<td>mw</td>
<td>mr</td>
</tr>
<tr>
<td>Adhesive</td>
<td>10363883</td>
<td>Locdite 406</td>
<td></td>
<td>0.8</td>
<td>3</td>
<td>Adhesion</td>
<td>Thrown Burned</td>
<td>Stable</td>
<td>COx, NOx</td>
<td>mw</td>
<td>ene, gw, acid, ozn</td>
</tr>
</tbody>
</table>
5 List of regulations

This list identifies international regulations for health, safety and environmental conditions.

The applicability status of the regulations will have to be defined on a case by case basis for each specific program and contract.

5.1 European Union directives

− The European Union directive (2002/96/EC), Waste from Electrical and Electronic Equipment (WEEE) valid from 2006, may be used as a guideline stating target values for Recovery, Recycling and Re-use.
− Registration, Evaluation and Authorization of Chemicals REACh (Directive CE n° 2006/121/CE) concerning the recording, the evaluation and the authorization of the chemical substances, as well as the restrictions applicable to these substances, instituting a European agency of the chemicals, modifying directive 1999/45/CE and abrogating the regulation (the EEC) N 793/93 of the Council and regulation (EC) N 1488/94 of the Commission as well as directive 76/769/CEE of the Council and directives 91/155/CEE, 93/67/CEE, 93/105/CE and 2000/21/CE of the Commission.
− PCB & PCT disposal (Directive CE n° 96/59/CE)
− Asbestos ban (Décret 1996-1133) for France
− Battery disposal (Directive CE n° 91/157/CEE)
− Ozone depleting substances (Directive CE nº 2037/2000)
− ROHS (Directive CE 2002/96)

5.2 Transportation regulations

Transportation of products recommended to be returned for safe disposal, needs to be approved by national/state-run Competent Authorities which issues transport certificates stating the components transportation classification based on the UN recommendation on the transport of Dangerous Goods ("orange book"). This international recommendation classifies dangerous goods by type of risk involved, UN classification code, and product category codes, UN Number.

The UN Number defines the packing group and packing instruction for the Product, maximum weight of the package and how the package shall be labeled for each mode of conveyance.

Note that the packing itself need to have type approval issued by a Competent Authority.

The recommendations need to be tailored for transportation by road, railway, sea and air.

− Agreement on Dangerous Goods by Road (Europe)
− Regulations Concerning the International Transport of Dangerous Goods by Rail (European law)
− Dangerous Goods Regulations IATA - international recommendation for air transports
− International Maritime Dangerous Goods (United Nations) for sea transports

The Transport certificates are often national and normally have a period of validity of a maximum of 10 years. At the time for disposal, new certificates must be applied for at the Competent Authorities concerned.

5.3 Support equipment and ground equipment

The PDF will be based upon the 2000/53/CE directive from the European Parliament and the council relating to Vehicles out of use.

This directive is valid only for the civil vehicles of a weight lower or equal to 3,5 ton. This directive will be used as a basis for the development of a future regulation which will include all the civil and military vehicles. This directive envisages certain measures and incentives so that
the phase of dismantling is taken into account from the start of the design of the equipment. It will particularly be based upon new standards, established according to scientific evolutions. The principal objective of this directive is the human, animal and environment protection, which will be done by better Vehicles out of use management, whose base will be the compilation of a disposal file which will allow providing the adequate solutions for the assumption of the waste responsibility or the use of less harmful replacement materials to the environment.

5.4 Navy equipment

The PDF should comply with the "green passport" requirements, whose principles are stated in the A.962(23) resolution from the International Maritime Organization (IMO) of December 5, 2003, supplemented by the A.980(24) resolutions and especially the A981(24) resolution of December 1, 2005, which is pressed on international texts into force such as the convention of Basle about trans-border transfer of waste.

The A.962(23) resolution is an equivalent of 2000/35/CE directive applicable to the maritime equipments of which one of the components is the "green passport", which is an instrument intended to facilitate the operations of dismantling and recycling of any ship that arrives at the end of its lifetime by the adoption of good practices all for the length of the life cycle of the ship (respectful of environment on the level of the dismantling site and making safe for the personnel carrying out this work), the objective being a progressive reduction of the use of any dangerous materials lasting the service life of the ship.

In order to have further information on the "green passport", we will refer to the NI 528 document published by Bureau Veritas.

The directives are intended to provide to the flag States, port States and re-user States, owners of ships, manufacturers of ships, suppliers of navy material and recycling installations, several indications on the "best practices", which take account of the ship recycling process throughout the ship life cycle.

They take account of certain codes and protocols relating to the environmental protection.

The list of the substances to be tracked is based on the directive 67/548/CEE of the European Parliament and the council, to which substances suggested by the states will be added intervening for the drafting of the convention.

5.5 Air equipment

No regulation or guidelines has yet been set up worldwide to dispose of aircrafts at the end of their service life.

However, the PDF could be compiled taking advantage of the PAMELA process: Process for Advanced Management of End of Life of Aircraft. This process has jointly been developed in 2005 by Airbus, SITA France, EADS JRC, EADS SOGERMA Services and the prefecture of the Hautes-Pyrénées in the frame of the LIFE ("L'Instrument Financier pour l'Environnement" = The Financial Instrument for the Environment) European environmental project.

This project makes it possible to characterize the best practices of aircrafts dismantling at the end of its lifetime in order to reach a rate of recycling of 85% from here 2008.

5.6 Ammunition

The PDF will comply with the STANAG 4518 edition 1 (2001) "Ammunitions disposal".

The goal of this NATO standardization agreement is to provide a common base to the NATO members on the principles and requirements of safety for the design and the evaluation procedures in order to guarantee the safety of the operations of elimination of ammunitions.
5.7 Nuclear waste

Each state member of the International Atomic Energy Agency has its own standards to manage nuclear waste.

However, the Agency released the three following documents (Safety Guides and information Circular) addressing nuclear wastes safety:

- Environmental and Source Monitoring for Purposes of Radiation Protection (RS-G-1.8),
- Management of Waste from the Use of Radioactive Materials in Medicine, Industry, Research, Agriculture and Education (WS-G-2.7),
- Code of Practice on the International Transboundary Movement of Radioactive Waste (INFCIRC/386)

Moreover, the Board of Governors approved a Safety Requirements publication, co-sponsored by the Organisation for Economic Co-operation and Development/Nuclear Energy Agency (OECD/NEA), on geological disposal (WS-R-4).

For further information, here is a list of major websites:

<table>
<thead>
<tr>
<th>Site address</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>France</strong></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.andra.fr/">http://www.andra.fr/</a></td>
<td>National radioactive waste management agency</td>
</tr>
<tr>
<td><a href="http://www.cea.fr/">http://www.cea.fr/</a></td>
<td>French government-funded technological research organisation</td>
</tr>
<tr>
<td><a href="http://www.asn.fr/">http://www.asn.fr/</a></td>
<td>The French Security Nuclear Authority</td>
</tr>
<tr>
<td><strong>USA</strong></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.wipp.energy.gov/">http://www.wipp.energy.gov/</a></td>
<td>Waste Isolation Pilot Plan</td>
</tr>
<tr>
<td><a href="http://www.energy.gov/">http://www.energy.gov/</a></td>
<td>USA Department of Energy</td>
</tr>
<tr>
<td><a href="http://www.nrc.gov/">http://www.nrc.gov/</a></td>
<td>USA Nuclear Regulatory Commission</td>
</tr>
<tr>
<td><a href="http://www.epa.gov/">http://www.epa.gov/</a></td>
<td>USA Environmental Protection Agency</td>
</tr>
<tr>
<td><strong>Canada</strong></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.nwmo.ca/">http://www.nwmo.ca/</a></td>
<td>Nuclear Waste Management Organisation</td>
</tr>
<tr>
<td><a href="http://www.nuclearsafety.gc.ca/">http://www.nuclearsafety.gc.ca/</a></td>
<td>Canadian Nuclear Safety Commission</td>
</tr>
<tr>
<td><strong>Germany</strong></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.bfs.de/">http://www.bfs.de/</a></td>
<td>Federal Office for radiation Protection</td>
</tr>
<tr>
<td><a href="http://www.dbe.de/">http://www.dbe.de/</a></td>
<td>Company for the Construction and Operation of Waste Repositories</td>
</tr>
<tr>
<td><strong>Belgium</strong></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.fanc.fgov.be/">http://www.fanc.fgov.be/</a></td>
<td>Federal Agency of Nuclear Control</td>
</tr>
</tbody>
</table>

Table 6 Websites for additional information concerning disposal
<table>
<thead>
<tr>
<th>Site address</th>
<th>Organization</th>
</tr>
</thead>
</table>
| Finland      | Nuclear Waste Management Expert  
http://www.posiva.fi/  
http://www.stuk.fi/  
Radiation and Nuclear Safety Authority  
Japan  
http://www.numo.or.jp/  
http://www.nisa.meti.go.jp/  
Nuclear Waste Management Organisation of Japan  
Nuclear and Industrial Safety Agency  
Switzerland  
http://www.nagra.ch/  
http://www.hsk.ch/  
National Cooperative for the Disposal of Radioactive Waste  
Division Principale de la Sécurité des installations nucléaires  
Sweden  
http://www skb.se/  
http://www.ski.se/  
Swedish Nuclear Fuel and Waste Management Co  
Swedish Radiation Safety Authority  
Europe  
http://www.nirond.be/  
ONDRAF: Organisme National des Déchets Radioactifs et des Matières Fissiles enrichies  
http://ec.europa.eu/  
European Commission about human and environment safety  
International  
http://www.nea.fr/  
Nuclear Energy Agency with 28 member states  
http://www.iaea.org/  
International Agency Energy Atomic with 144 member states  
http://www.unscear.org/  
United Nations Scientific Committee on the effects of atomic radiation  

End of data module
Chapter 18

Interrelations to other ASD specifications

Table of Contents

Interrelations to other ASD specifications........................................................................................................... 1
References......................................................................................................................................................... 1
1 General ..................................................................................................................................................... 2
1.1 Introduction ............................................................................................................................................. 2
1.2 Objective ................................................................................................................................................ 2
1.3 Scope .................................................................................................................................................... 2
2 Benefits of Using the ASD specifications suite ......................................................................................... 2
2.1 Conceptual background of ASD specification ...................................................................................... 2
2.2 Integrated concept of operation ........................................................................................................... 4
3 Interrelation to S1000D ........................................................................................................................ 4
3.1 Purpose of S1000D .............................................................................................................................. 4
3.2 S3000L/S1000D .................................................................................................................................. 4
4 Interrelation to S2000M ........................................................................................................................... 5
4.1 Purpose of S2000M .............................................................................................................................. 5
4.2 S3000L/S2000M .................................................................................................................................. 5
5 Interrelation to S4000M ........................................................................................................................... 6
5.1 Purpose of S4000M .............................................................................................................................. 6
5.2 S3000L/S4000M .................................................................................................................................. 6
6 Interrelation to S5000F ........................................................................................................................... 8
6.1 Purpose of S5000F .............................................................................................................................. 8
6.2 S3000L/S5000F .................................................................................................................................. 8

List of Tables

1 References ..................................................................................................................................................... 1

List of figures

1 Acquisition logistics main business processes (1) ...................................................................................... 3
2 Acquisition logistics main business processes (2) ...................................................................................... 4
3 Process overview from SMA tasks via LSA tasks to maintenance packages........................................... 7
4 Traceability of scheduled maintenance ................................................................................................... 8

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 1</td>
<td>Introduction to the specification</td>
</tr>
<tr>
<td>DEX1A&amp;D</td>
<td>Aerospace and Defense Product Breakdown for Support</td>
</tr>
<tr>
<td>DEX3A&amp;D</td>
<td>Aerospace and Defense Task Set</td>
</tr>
<tr>
<td>S1000D</td>
<td>International specification for technical publications using a common source database</td>
</tr>
</tbody>
</table>

Effectivity: All

S3000L-A-18-00-0000-00A-040A-A

Chap 18
1 General

1.1 Introduction
The S3000L should not be considered as a stand alone entity. In the environment of logistics, there are existing AeroSpace and Defence Industries Association of Europe (ASD) specifications which describe the development of technical documentation (S1000D) and define the materiel management processes and procedures to be used in support of a product (S2000M). Also the new specification S4000M concerning Scheduled Maintenance Analysis (SMA) fits perfectly into the complete specification suite. The future specifications concerning the handling of in-service data (S5000F) and a super ordinate specification concerning the Integrated Logistic Support Management (SX000I) will complete the specification suite and then will cover the most important areas of product supportability.

1.2 Objective
This chapter provides an overview of how the existing ASD specifications and the S3000L will be harmonized. It explains where the connecting point are and what benefits can be achieved by using the S3000L together with the other existing and well proven ASD specifications S1000D, S2000M and the new specification S4000M.

1.3 Scope
This chapter is directed to logistics personnel who organize logistic analysis activities like an Integrated Logistics Support (ILS) manager or a Logistic Support Analysis (LSA) manager. The interdisciplinary character of the logistic supportability should be highlighted by this chapter.

2 Benefits of using the ASD specifications suite

2.1 Conceptual background of ASD specification
The underlying basic concept for the development of a suite of ASD specifications in order to support the acquisition logistics main business processes was developed in early 1993.

During an international workshop in Paris, with high-level industrial and government participation, the major domains of logistic support during an acquisition phase (generally for newly built systems and major retrofit campaigns) were identified and networked. These domains provide the framework for future specification development activities with the goal of covering the complete product life cycle.

These domains were:
- Logistic support analysis
- Provisioning
- Order administration
- Technical documentation
At the time of the ASD workshop in Paris, mentioned above, the technical documentation domain was already covered by S1000D and the provisioning and order administration domains were covered by S2000M.

No ASD specification existed, however, for the LSA at that time. In a first effort to close this gap, the ASD decided to develop a new specification, S4000M - International procedure handbook for the development of scheduled maintenance programs for military aircrafts. Then in 2005, the ASD together with the Aerospace Industries Association of America (AIA) started the development of S3000L - International procedure specification for Logistics Support Analysis. Details thereof are provided in Chap 1 - Introduction to the specification, and are therefore not repeated here.

In the meantime it was obvious that another specification was needed to satisfy the requirements for the support of the full product life cycle. The functionality will concentrate on maintenance and operational data feedback and the specification will be known as S5000F.

The development of S5000F commenced in October, 2008. Estimated time for completion is 15 months.
The future network will comprise a total of five specifications which will cover the major support functions of the product life cycle.

### 2.2 Integrated concept of operation

The individual ASD specifications should not be considered as stand-alone entities. In the environment of acquisition logistics they should be used as a coherent set (suite) of functional specifications with common data definitions and integrated data exchange. This integrated concept will ensure that all major functionalities of acquisition logistics are efficiently covered without overlap. The specifications are internationally accepted and are broadly used in the aerospace and defence business, but are not limited to these areas.

Of all the ASD specifications, S3000L plays a leading role. It is the principal tool to:

- design the products relevant to maintainability, reliability, testability and to optimize Life Cycle Cost (LCC).
- define all required resources to support the product in its intended use during inservice operation.

### 3 Interrelation to S1000D

#### 3.1 Purpose of S1000D

S1000D was produced to establish an international specification for procurement, production and distribution of technical documentation and learning content. This specification can be applied to the documentation of any type of equipment including both military and civil products.

**Note**

Since 2007, ASD, AIA and Air Transport Association of America (ATA) will jointly further develop, maintain and promote S1000D in the international arena.

#### 3.2 S3000L/S1000D

The task information developed during the LSA process is the baseline for the maintenance procedures to be produced in accordance with S1000D. The LSA data is also the input for the maintenance planning information.
The task information includes, for example:

- Task description
- Preliminary requirements
  - Required conditions
  - Required persons
  - Required equipment
  - Safety conditions

This data must be used as a direct input to the corresponding procedure in the maintenance publications. Depending on the publication production environment, some of this data can be directly included in the procedure. During the update of the procedure, the author can be notified of any changes.

If a full set of maintenance planning data is developed in the LSA process this data can be transferred directly to the maintenance planning publication data modules.

The details of the data set to be exchanged are given in DEX1A&D and DEX3A&D and the specification S1003X - S1000D and S3000L interface specification.

Note
There is not always a one-to-one relation between a task coming from the LSA process and a procedural data module, since tasks can be grouped into one procedural data module or split into several reusable procedural data modules.

4 Interrelation to S2000M

4.1 Purpose of S2000M

S2000M originally defined the materiel management processes and procedures to be used in support of aircraft and other aerospace airborne and ground equipment supplied to military customers. S2000M has now been revised to include the business processes and data applicable to any military product. Although the S2000M was designed for military product support, it may nevertheless be used for the support of any other complex Product.

The processes described within S2000M cover the interfaces between the Contractor and the Customer, which, when based upon contractual agreements, will provide the typical deliverables of the logistic materiel management:

- Provisioning
- NATO codification (as a special military feature)
- Procurement planning
- Order administration
- Invoicing
- Repair administration
- S2000M Light

4.2 S3000L/S2000M

During the S3000L LSA process, information will be generated that will determine the range and depth of the maintenance of the product, as well as the required material resources during in-service operation. On the S2000M side, the appropriate functionality for the interface to S3000L is the provisioning functionality. Provisioning, according to chapter 1 of S2000M, is the process of selecting support items and spares necessary for the support of all categories of products.

The compilation of provisioning data for those items identified as relevant for a customer's maintenance and support concept is the prime objective of that phase.

Provisioning data is compiled according to compilation rules specified in S2000M. These rules are normally confirmed or modified/specified during a Guidance Conference (GC). The GC
takes place at the start of any project in which the S2000M procedures are to be operated. During the GC, it will be specified in detail as to what extent LSA data can/shall be used to support the provisioning process. Assuming that a full LSA program is authorized for a product then, as an example, the following S2000M Chapter 1 data elements can be taken/derived from LSA:

- Authorized life (AUL)
- Essentiability code (ESC)
- Integrated logistic support number (ILS)
- Maintenance percent (MAP)
- Mean time between failures (TBF)
- Part number (PNR)
- Scrap rate (SRA)
- Shelf life code (SLC)
- Size of packaged unit (SPU)
- Size of unpackaged unit (SUU)
- Source maintenance recoverability code (SMR)
- Special storage (STR)
- Time between overhauls (TBO)
- Time between scheduled shop visits (TSV)
- Weight of packaged unit (WPU)
- Weight of unpackaged unit (WUU)

Note
The data being derived from LSA need not be limited to the list above. There can be more common data between the S2000M and S3000L areas. It should be highlighted that especially data which are common between S2000M and S3000L (simple example: the name of a part/equipment) should be harmonized as much as possible. The leading discipline for technical values should be identified during the GC.

5 Interrelation to S4000M

5.1 Purpose of S4000M
S4000M provides a means for developing the scheduled maintenance tasks and intervals which will be acceptable to operators, manufacturers and regulatory authorities (if applicable). The scheduled maintenance task and interval details will be developed by coordination with specialists from operators, manufacturers and the regulatory authority. Specifically, S4000M outlines the general organization and decision processes for determining scheduled maintenance requirements initially projected for the life of the product.

5.2 S3000L/S4000M
LSA and SMA are interconnected very closely. Only the common view on unscheduled maintenance and scheduled or preventive maintenance respectively gives a complete impression of maintenance activities. The identification of the requirements for any scheduled task is the same as for the unscheduled tasks. For that reason, the documentation of the tasks in the LSA database can be done in almost the same way (only the justifying event differs). Also the packaging of the scheduled maintenance activities can be supported by the use of the LSA database. The scheduled tasks for each relevant equipment or subsystem can be documented in the LSA database as well as in the final relevant inspection and overhaul packages. Additionally, the individual tasks and the packages can be linked together for traceability purposes, as well as for reporting and evaluation purposes.

The interconnecting data is primarily for the identification of the scheduled task itself (the task will be documented in the LSA database) and the related interval/threshold. Interval adoption resulting from the scheduled maintenance packaging requirements will also be documented in the LSA database.
In the following figure, the entire process of identifying SMA tasks within the applied SMA method to a final harmonization of scheduled maintenance in form of inspection and overhaul packages.

**ASD S4000M - Scheduled Maintenance Analysis**

<table>
<thead>
<tr>
<th>System and Powerplan Analysis</th>
<th>Structure Analysis</th>
<th>Zonal Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifies tasks</td>
<td>identifies tasks</td>
<td>identifies tasks</td>
</tr>
<tr>
<td>applicable and effective</td>
<td>applicable and</td>
<td>applicable and effective</td>
</tr>
<tr>
<td>scheduled or preventive</td>
<td>scheduled or</td>
<td>scheduled or preventive</td>
</tr>
<tr>
<td>maintenance actions</td>
<td>preventive</td>
<td>maintenance actions</td>
</tr>
</tbody>
</table>

**ASD S3000L - Logistic Support Analysis (Logistic database)**

Fig 3  Process overview from SMA tasks via LSA tasks to maintenance packages

Rules for the documentation of scheduled maintenance packages within a LSA database must be established and harmonized with the customer during the LSA GC. It is strictly recommended that the validity of tasks is clarified within the LSA database.

The LSA tasks derived from the original SMA can serve as the documentation of SMA results, too. It should be possible for a S3000L application to link corresponding SMA documents. Also, the need for modification of task thresholds or intervals, or even the need for design changes, can be documented in the LSA as well as the original situation in the SMA tasks. Traceability should be the main driver for the documentation of SMA within the logisticians’ environment, as shown in Fig 4.
6 Interrelation to S5000F

6.1 Purpose of S5000F
S5000F provides a means for the feedback of in-service information to those functional domains/activities from which it was originally generated and later transmitted to the customer to support operations (refer to Fig 2).

6.2 S3000L/S5000F
In-service reality and the maintenance concept/tasks and the logistic support requirements developed by an S3000L LSA process must be continuously compared to ensure the identification of required revaluation or adaptation. During operation, it becomes evident whether the predictions of the logistic analysis processes can bear comparison with the reality of the day-to-day experiences of the system operators. In-service data must be collected carefully and compared to the existing maintenance situation for permanent optimization of logistic support.

Specific data elements will be defined during the development of S5000F. At this time only the information clusters are addressed as they will be covered in S5000F:

- data for the analysis of defects, events and system/component health
- data for usage optimization
- data for integrated fleet management
- data relevant to consumption
- data relevant to Engineering Record Cards,
- data to support the management of Performance Based Logistics contracts
- data to support LCC considerations
- data to support product liability issues

End of data module
Chapter 19

Data elements

Table of contents

Data elements ..................................................................................................................................... 1
References ....................................................................................................................................... 4
1 General ......................................................................................................................................... 5
1.1 Introduction .............................................................................................................................. 5
1.2 Objective .................................................................................................................................. 5
1.3 Scope ........................................................................................................................................ 5
1.4 Out of scope ............................................................................................................................. 5
2 Overview ...................................................................................................................................... 5
2.1 Unified modeling language .................................................................................................... 5
2.1.1 UML Class model ............................................................................................................... 6
2.1.2 Special guidance for reading the S3000L data model ....................................................... 12
2.2 Organization of this chapter ................................................................................................. 13
2.3 S3000L data types ................................................................................................................ 14
2.3.1 Overall description ........................................................................................................... 14
2.3.2 Graphical representation ................................................................................................. 14
2.3.3 Data type Date .................................................................................................................. 15
2.3.4 Data type Id ...................................................................................................................... 15
2.3.5 Data type Descr ............................................................................................................... 16
2.3.6 Data type Class ............................................................................................................... 16
2.3.7 Data type Classification ................................................................................................. 17
2.3.8 Data type Prp ................................................................................................................... 17
2.4 Unknown values ................................................................................................................... 19
3 Model overview .................................................................................................................... 19
4 Functional areas .................................................................................................................... 20
4.1 UoF Project ............................................................................................................................ 20
4.1.1 Overall description ......................................................................................................... 20
4.1.2 Graphical representation ............................................................................................... 21
4.1.3 UoF Project - New class and interface definitions ......................................................... 21
4.2 UoF Product Usage .............................................................................................................. 24
4.2.1 Overall description ......................................................................................................... 24
4.2.2 Graphical representation ............................................................................................... 25
4.2.3 UoF Product Usage - New class and interface definitions ........................................... 25
4.2.4 UoF Product Usage - Additions to referenced class and interface definitions ............. 27
4.3 UoF Breakdown Structure ................................................................................................. 27
4.3.1 Overall description ......................................................................................................... 27
4.3.2 Graphical representation ............................................................................................... 29
4.3.3 UoF Breakdown Structure - New class and interface definitions ................................... 29
4.3.4 UoF Breakdown Structure - Additions to referenced class and interface definitions .... 32
4.4 UoF Part .................................................................................................................................. 32
4.4.1 Overall description ......................................................................................................... 32
4.4.2 Graphical representation ............................................................................................... 34
4.4.3 UoF Part - New class and interface definitions ............................................................. 34
4.4.4 UoF Part - Additions to referenced class and interface definitions ............................... 40
4.5 UoF Breakdown Element Realization ................................................................................ 40
4.5.1 Overall description ......................................................................................................... 40
4.5.2 Graphical representation ............................................................................................... 41
4.5.3 UoF Breakdown Element Realization - New class and interface definitions .............. 41

Applicable to: All
S3000L-A-19-00-0000-00A-040A-A
Chap 19

DMC-S3000L-A-19-00-0000-00A-040A-A_001_00_EN-US.doc

2010-04-01 Page 1
List of tables

1 References ............................................................................................................. 4
2 Association cardinalities used in S3000L ............................................................... 7

List of figures

1 Example of a class in UML .................................................................................... 6
2 Example of relational representation of the UML class given above ..................... 7
3 Example of an association between classes in UML ............................................. 7
4 Example of relational representation of the project and contract association given above ................................................................. 8
5 Example of an association class in UML .................................................................. 8
6 Example on relational representation of the association class attributes given above ........................................................................... 8
7 Example of a directed association in UML ............................................................. 9
8 Example of specialization (inheritance) relationships between classes in UML ........ 10
9 Example of relational representation of the specializations given above ............. 10
10 Example of a composition aggregation in UML ................................................... 11
11 Example on relational representation of the composition aggregation given above .................................................................................................................................. 11
12 Example of Interface and Realize relationships in UML ....................................... 12
13 Example on relational representation of interface definition .................................. 12
References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 12</td>
<td>Maintenance task analysis</td>
</tr>
<tr>
<td>Chap 20</td>
<td>Data exchange</td>
</tr>
<tr>
<td>Chap 22</td>
<td>Data element list</td>
</tr>
<tr>
<td>GEIA-STD-0007</td>
<td>Logistics Product Data</td>
</tr>
<tr>
<td>ISO 10303:239 PLCS</td>
<td>Product Life Cycle Support</td>
</tr>
<tr>
<td><a href="http://www.pdm-if.org/pdm_schema">www.pdm-if.org/pdm_schema</a></td>
<td>PDM Schema usage guide</td>
</tr>
<tr>
<td><a href="http://www.oasis-open.org">www.oasis-open.org</a></td>
<td>Organization for the Advancement of Structured Information Standards</td>
</tr>
<tr>
<td><a href="http://www.uml.org">www.uml.org</a></td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td><a href="http://www.plcs-resources.org">www.plcs-resources.org</a></td>
<td>Product Life Cycle Support website</td>
</tr>
</tbody>
</table>
1 General

1.1 Introduction
This chapter defines a coherent data model for the data that can be exchanged between the Logistics Support Analysis (LSA) process as defined in S300L, and related business processes. These related business processes can be either business processes that are dependent upon data being produced as a result of the LSA, or business processes that provide input to the LSA process.

The data model is described using the UML (Unified Modeling Language) version 2, class model (www.uml.org).

Each attribute defined in the UML class model is defined in Chap 22, Data element list.

The data model has been influenced by ISO 10303-239 Product Life Cycle Support (PLCS), in order to simplify the use of PLCS Data EXchange specifications (DEX) for actual data exchanges (refer to Chap 20).

The data model defined in this chapter has been defined during a series of workshops. Some of these workshops have focused on data required to support the business process described in S300L, whereas other workshops have focused on getting a good integration and harmonization between S300L and the task related schemas in S1000D.

1.2 Objective
The objective for this chapter is to define a coherent data model for the data that can be exchanged between the LSA process, as defined in S300L, and related business processes.

1.3 Scope
The following areas are within scope of the data model:

- Definition of the LSA project and the products that is to be supported
- Support the early phases of the LSA project in terms of selecting the LSA candidate items as well as selecting analysis activities that is to be performed for each candidate item
- Support the LSA FMEA (Failure Mode and Effect Analysis)
- Document the results from the maintenance and operational task analysis activities

1.4 Out of scope
The data model has been defined to support one project at the time. Any implementation of the data model in a software package that will support multiple projects needs to define data which can be reused between projects, and implement an applicability model to support this.

The data model does not provide full support for all the identified candidate item analysis activities, but focuses on Maintenance Task Analysis (MTA) and LSA FMEA. Results from other analysis activities, eg LORA, can be referenced by summarized descriptions or by explicit document references.

The data model does not cover all data elements supporting the LSA process, but just those data elements that typically will be exchanged between a contractor and his subcontractors, partners and customers.

2 Overview

2.1 Unified modeling language
The Unified Modeling Language™ - UML - is a widely used technique to model not only application structure, behavior, and architecture, but also business processes and data structures.
UML consists of a set of different modeling techniques of which this chapter only uses one, namely the UML class model. A class model defines a static view of the information (classes, attributes and relationships) that is needed to support the business processes.

2.1.1 UML Class model

Class models are the most widely used part of UML. Class models shows the things that are to be represented, and their relationships.

This section gives a short overview of the UML constructs that are used in the S3000L data model. It is not a complete description of the UML class model concepts. Each UML class model concept is also translated into a relational table example. These relational table examples are provided for those readers that has an understanding of relational databases, but no previous knowledge of UML. The translations between UML and relational tables shall be seen as examples on how UML class model concepts can be represented using a relational database.

2.1.1.1 Class

The rectangle in a class diagram is called a classifier. The classifier gives you the name of the class together with an enumeration of its attributes. Each attribute contains the attribute name, data type and cardinality.

Note

A complete description of a class also includes its behavior (also known as methods), but this is out of scope of the S3000L data model.

An example of a class is given in the figure below. It shows a project class with two attributes, Project_identifier and Project_name. Each attribute also shows its data type and cardinality.

A class can have zero, one or many instances. Example on a project instance is the New Fighter Aircraft project.

Note

A class in UML can be viewed as a table in a relational database, and an instance of that class can be seen as a row within that table. The attributes of a class can be seen as the table columns.
Fig 2 Example of relational representation of the UML class given above

**Note**

The relational table column that constitutes its primary key, is emphasized by underlining the column name(s).

A class can also have relationships to other classes in terms of association, composition, aggregation, generalization/specialization and realization.

2.1.1.2 Association

Associations represent interdependencies between classes. Associations are often viewed as just another attribute where the attribute is modeled as a class with a connector between the containing class and the class representing the attribute.

An example of an association is shown in the figure below. It shows that a project has at least one associated contract, and that a contract can be associated with zero, one or many instances of project. The cardinality for the association is given at the respective end of the association.

<table>
<thead>
<tr>
<th>Association</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One (and only one) instance of the associated class must be associated with each instance of the associating class (a mandatory association)</td>
</tr>
<tr>
<td>0..*</td>
<td>Zero, one or many instances of the associated class can be associated with each instance of the associating class (an optional association)</td>
</tr>
<tr>
<td>1..*</td>
<td>At least one instance of the associated class must be associated with each instance of the associating class (a mandatory association)</td>
</tr>
</tbody>
</table>
Note
An association in UML can be viewed as a relationship table in a relational database, where the columns in the table represent the primary keys from the respective class, and the table rows represent references to the associated instances.

<table>
<thead>
<tr>
<th>PROJECT_CONTRACT_RELATIONSHIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECT_NAME</td>
</tr>
<tr>
<td>New Fighter Aircraft</td>
</tr>
<tr>
<td>New Fighter Aircraft</td>
</tr>
</tbody>
</table>

ICN-B6865-S3000L0203-001-01

Fig 4 Example of relational representation of the project and contract association given above

2.1.1.3 Association class
An association may also have an association class. An association class adds information to the association.

An example of an association class is given in the figure below. It shows that a contract can be associated with other contracts, and that each association can contain information on the type of association that is established between the two contracts. This information is defined within the association class Contract_relationship by using the Contract_relationship_type attribute.

ICN-B6865-S3000L0204-001-01

Fig 5 Example of an association class in UML

Note
An association class in UML can be viewed as adding columns to a relationship table in a relational database.

<table>
<thead>
<tr>
<th>CONTRACT_CONTRACT_RELATIONSHIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTRACT_ID_1</td>
</tr>
<tr>
<td>CCT-2008-011</td>
</tr>
</tbody>
</table>

ICN-B6865-S3000L0205-001-01

Fig 6 Example on relational representation of the association class attributes given above

2.1.1.4 Directed association
A directed association has an open arrowhead at the target end of the association. In a directed association, two classes are related, but only one class knows that the relationship exists.

An example of a directed association is the possibility to associate an higher failure mode effect with an LSA candidate that represents the higher level function, including the end item.
However, there is no requirement for the LSA candidate to be able to navigate to any associated higher failure mode effects.

![Example UML directed association](image)

**Fig 7 Example of a directed association in UML**

**Note**

A directed association in UML can be viewed as a relationship table in a relational database, in the same way as described for an association above.

2.1.1.5 Generalization and specialization

Specialization refers to an ‘is-a’ relationship and is modeled by a solid line connector with a hollow triangle at one end. The triangle points to the parent (ie the generalized class). Specialization is often referred to as inheritance.

The most important aspect of the generalization/specialization relationship is that the child class gets all the features of the parent class, and can then add features of its own, ie a child class inherits all the attributes and relationships defined for its parent class.

Another important aspect of specialization is substitutability. Substitutability means that wherever a parent class is being used, a child class can be substituted for its parent.

An example of the specialization/generalization association is Part and its specialization into Hardware_part and Software_part, respectively. In the example given below, the Part class has the defined attributes Part_identifier and Part_name. This means that both the Hardware_part and Software_part classes also has the Part_identifier and Part_name attributes, even though they are not explicitly enumerated in the attribute list for the respective class. The respective class also defines additional attributes which are unique for its respective special characteristics.
Fig 8  Example of specialization (inheritance) relationships between classes in UML

Note
A special characteristic that can be defined for the parent class is that it can be defined as an ‘abstract’ class. This means that the parent class itself cannot be instantiated, but must be substituted by any of its child classes. This is denoted in the model by making the class name (classifier) italic. Part in the example above is defined as an abstract class.

Specialization can be viewed as multiple tables with the same initial set of columns (incl. the primary key).

<table>
<thead>
<tr>
<th>HARDWARE_PART</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART_ID</td>
</tr>
<tr>
<td>240-45-656654</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SOFTWARE_PART</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART_ID</td>
</tr>
<tr>
<td>190-23.143244</td>
</tr>
</tbody>
</table>

Fig 9  Example of relational representation of the specializations given above

Note
Both the Actual_software_size value and Actual_software_size UoM (Unit of Measure) are included in the Prp data type.

2.1.1.6 Aggregation
Aggregation defines whole and part relationships, ie it indicates that one class is part of another class. In an aggregation relationship, the child class instance can outlive its parent class.

Note
An example of an aggregation relationship is Car and its Wheels, where the Car represents the whole class and the Wheels represents parts of the overall Car. However, the Wheel class instances can live independently of the Car class instance.

Note
Aggregation relationship is currently not used in the S3000L data model, but is described as background information for the composition aggregation defined in the next paragraph.
2.1.1.7 Composition aggregation

Composition aggregation is just another form of aggregation relationship, where the child class’s instance lifecycle is dependent on the parent class’s instance lifecycle.

An example of a composition aggregation is the possibility to define that a Task consists of one or many Subtasks.

![Example of composition aggregation in UML](image)

**Fig 10** Example of a composition aggregation in UML

**Note**

Composition aggregation can be viewed as two relational tables, where the relational table that represents the aggregated class has a primary key which initial columns corresponds to the primary key for the table that represents the composition class.

<table>
<thead>
<tr>
<th>TASK</th>
<th>SUBTASK</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK_ID</td>
<td>TASK_NAME</td>
</tr>
<tr>
<td>TSK-1112</td>
<td>Remove engine</td>
</tr>
<tr>
<td>SUBTASK_ID</td>
<td>SUBTASK_ROLE</td>
</tr>
<tr>
<td>TSK-1112</td>
<td>STSK-1</td>
</tr>
<tr>
<td>TSK-1112</td>
<td>STSK-2</td>
</tr>
</tbody>
</table>

**Fig 11** Example on relational representation of the composition aggregation given above

2.1.1.8 Interface and the realize relationship

Interfaces are a way of modeling features that are common for a set of classes.

Any class that realizes (implements) an interface must support the features defined by the interface. These features include attributes, relationships and behavior.

An example of interface in the S3000L data model is the LSA_candidate interface. This interface is realized by both Breakdown_element_revision and Part. This means that the attributes and relationships that are defined for the LSA_candidate interface must be supported by both Part and Breakdown_element_revision.

**Note**

Interfaces simplify the data model, e.g., a similar relationship between one class and a set of other classes can be directed towards an interface which in turn is implemented by each class that can be part of that relationship.
The graphical representation for the realization connector is almost identical to the generalization connector, except that the connector is a dashed line instead of a solid line.

![Example UML interface](image)

**Fig 12 Example of Interface and Realize relationships in UML**

**Note**

Interface definition can be viewed as adding columns to existing relational tables (see example below) as well as adding relationship tables that will represent the interface associations.

<table>
<thead>
<tr>
<th>BREAKDOWN_ELEMENT_REVISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE_ID</td>
</tr>
<tr>
<td>190-23-143244</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PART</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART_ID</td>
</tr>
<tr>
<td>240-45-656654</td>
</tr>
</tbody>
</table>

**Fig 13 Example on relational representation of interface definition**

### 2.1.2 Special guidance for reading the S3000L data model

#### 2.1.2.1 Classes and attributes

The data types that are used in the S3000L class model, are based on basic ISO 10303:239 Product Life Cycle Support constructs, and OASIS usage guidance on how to implement PLCS. For further information, refer to Para 3, S3000L data types.

Each attribute used in the S3000L data model are defined in Chap 22 Data element list.

#### 2.1.2.2 Graphics

The graphical diagrams contain classes that both filled and not. A filled class represents classes that are introduced and defined within the Unit of Functionality (UoF) under consideration. Classes without any fill (ie white) are classes that are defined in another UoF, and are presented without any of its attributes.
2.1.2.3 Textual descriptions
The use of the terms "specific" and "instance of" in the textual descriptions means the same thing, i.e., it refers to a specific instance of a class. For a description of what is meant by an instance, refer to Para 2.1.1.1 above.

2.2 Organization of this chapter
This chapter is organized into a set of Units of Functionalities (UoF). Each UoF divides the overall data model into a set of smaller data models, which defines classes and attributes required to document a specific aspect of the LSA.

Note
The set of UoF does not follow the chapters of S3000L, but is structured in accordance with the order by which the data typically is created during LSA process.

Each UoF contains:
- An overall description of the UoF
- A graphical representation of the class model for the UoF
- Definitions of new classes and interfaces identified within the UoF
- Additions to classes and interfaces defined within other UoFs

Each class defined in the data model has:
- A short description of its intent
- An enumeration of its attributes, including attribute cardinality
  - Where no cardinality is given, the attribute is mandatory, and only one value can be given
  - "zero or one", means that the attribute is optional
  - "one or many", means that the attribute is mandatory and that it can have multiple values
  - "zero, one or many", means that the attribute is optional, but it can also have more than one value
- An enumeration of its associations
  - Associations described as "can be" means that the association is optional
  - Associations described as "must be" means that the association is mandatory
- An enumeration of implemented interfaces
  - Implemented interfaces means that the class under consideration must support all the attributes and associations defined by the interface
- Special recommendations, eg
  - Special recommendations regarding the use of the attribute data types
  - Special recommendations regarding use of applicability

Note
Definitions for the attributes used in the data model are provided in Chap 22 Data element list, and are not repeated in this paragraph.

Note
Any reference to GEIA-STD-0007 is provided for informational purposes only, based on the fact that GEIA-STD-0007 (former DoD-STD-1388-2b) is widely known in the LSA community.
Note
Where no special recommendation is stated for an attribute of type Prp (property), the recommendation is to use the Value_with_unit_property class for the representation of its value.

2.3 S3000L data types
2.3.1 Overall description
Data types used in the S3000L data model are not the same as usually encountered in data models, eg integer, real, string. In order to create a richer data model, S3000L has defined a set of data types that are closely related to the basic OASIS PLCS templates (www.oasis-open.org). These templates can be seen as small instantiation patterns, which ensure that each basic data element has an associated set of metadata.

For example, consider part identification (part number). In a traditional data model a part number would typically be represented as a string value. There is often, for example, no additional information about the organization that provided the part number.

In the OASIS PLCS template for assigning_identifier, the string value representing the part number would always be defined together with the following attributes:

- Information about the type of part number that is being represented (eg NATO Stock Number)
- Identification of the organization that determined ("owns") the part number
- Information on how the organization is identified (eg NCAGE code).

The following data types have been used throughout the S3000L data model:

- Date
- Id (Identifier)
- Descr (Description)
- Class
- Classification
- Prp (Property)

2.3.2 Graphical representation
2.3.3 Data type Date

The Date data type is used to represent calendar dates.

Date data type attributes:

- Year\_component
- Month\_component
- Day\_component

**Note**

All Date attributes are defined as integers. The rules for how to populate the respective value are defined in the data elements list.

**Note**

The Date data type does not include any format, eg 23/10/08 or 2008-10-23. Date format is avoided in order to simplify data exchange.

2.3.4 Data type Id

The Id data type is used to represent any form of identification, including names.

Id data type attributes:

- Identifier
- Identifier_class
- Identifier_set_by_organization (zero or one)

The Identifier attribute contains the identifier as such.

The Identifier_class attribute determines the type of identifier that is provided.

The Identifier_set_by_organization identifies the organization that determined ("owns") the identifier.

Note
The Identifier_set_by_organization relates to an Organization as the data type. The Organization class is defined in UoF Project, and contains an Organization_identifier (optional) and an Organization_name, which are both of data type Id. This means that the identification and/or name of the Organization comes with a classification of what type of identification is being used to identify the Organization (eg NCAGE code) and an optional determination of who determined ("owns") the Organization identifier.

2.3.5 Data type Descr

The Descr data type is used to represent any form of textual description (free form).

Descr data type attributes.
- Description
- Description_provided_by_organization (zero or one)
- Description_provided_date (zero or one)

The description attribute contains text that provides further information about the subject under consideration.

Note
The Description attribute must allow for descriptions in an indentured manner.

The Description_provided_by_organization identifies the organization that provided the description. The Description_provided_date identifies the date when the description was provided.

Note
The Descr attributes Description_provided_by_organization and Description_provided_date enables multiple descriptions to be provided for different purposes for the same object, and still be distinguished by organization and date.

2.3.6 Data type Class

The Class data type is used for representing references to terms, which can be used for classifications. The data model assumes that each term used has a definition in an external Reference Data Library (RDL). An RDL can be regarded as a glossary containing terms and definitions. For more information on the use of an RDL refer to www.plcs-resources.org.

Class data type attributes:
- Class_name
- Class_rdl (zero or one)

The Class_name attribute contains the name of the term being used for classification.

The Class_rdl attribute contains a reference to the source (RDL), where the definition of the term can be found.
2.3.7 Data type Classification

The Classification data type is used for representing dated assignments of classes, ie the date when the classification is done is as important as the classification itself.

Classification data type attributes:
- Classification_date

The Classification data type has the following associations:
- An association with the Class being used for dated classification

2.3.8 Data type Prp

2.3.8.1 Prp

The Prp data type is used for representing property values.

Note
The property value itself is represented using one of the subclasses of Property_representation.

Prp associations:
- An association with one or many instances of Property_representation which contains the actual value(s) for the property.

2.3.8.2 Property_representation

The Prp class has a one-to-many association with Property_representation. The reason for this is that the same property may have multiple representations, eg a length property may be represented in both inches and centimeters.

Note
Multiple representations are not the same as having multiple values for an attribute. Multiple representations just mean that the same value can be represented in multiple ways.

Each Property_representation can also include information on how and when a property value was set. The method by which the value has been determined (eg estimated, calculated) can be defined using the Property_determination attribute. When the value was set can be defined using the Property_creation_date attribute.

Property_representation attributes:
- Property_determination (zero or one)
- Property_creation_date (zero or one)

Property_representation associations
- An association with the Prp for which it represents a value

Note
The Property_representation class is an abstract class, ie any of its subclasses must be used for the actual representation of the value of a property.

Property_representation subclasses:
- Value_with_unit_property (a property with a value and a unit of measure)
- Value_with_limit_property (a property with a value and a unit of measure, where the value is defined as being either a "minimum" or "maximum" value)
- Value_with_tolerances_property (a property with a value and unit of measure along with acceptable upper and lower offset values)
− Value_range_property (a property with no fixed value, but defined by upper and lower limits along with a unit of measure)
− Text_property (a property value described as a string, eg "Green")

**Note**

Where no explicit recommendations have been made for attributes that are of data type Prp, in the S3000L data model below, the default is that they use the Value_with_unit_property representation, ie the property value is provided as a value and a unit of measure.

2.3.8.3 Value_with_unit_property

The Value_with_unit_property class is used to represent a value with its unit of measure.

The Value_with_unit_property class is a specialization of the Property_representation class.

**Note**

Value_with_unit_property is the default representation for all attributes that are of data type Prp.

Value_with_unit_property attributes:

− Property_determination (inherited from class Property_representation)
− Property_creation_date (inherited from class Property_representation)
− Value
− Unit

2.3.8.4 Value_with_limit_property

The Value_with_limit_property class is used to represent a value with its unit of measure, where the value is defined as being either a "minimum" or "maximum" value.

The Value_with_limit_property class is a specialization of the Value_with_unit_property class.

Value_with_limit_property attributes:

− Property_determination (inherited from class Property_representation)
− Property_creation_date (inherited from class Property_representation)
− Value
− Unit
− Limit_qualifier

2.3.8.5 Value_with_tolerances_property

The Value_with_tolerances_property class is used to represent a value with its unit of measure, along with acceptable upper and lower offset values.

The Value_with_tolerances_property class is a specialization of the Value_with_unit_property class.

Value_with_tolerances_property attributes:

− Property_determination (inherited from class Property_representation)
− Property_creation_date (inherited from class Property_representation)
− Value
− Unit
− Upper_offset_value
− Lower_offset_value

2.3.8.6 Value_range_property

The Value_range_property class is used to represent a property with a value range in terms of upper and lower limits along with a unit of measure.
The Value_range_property class is a specialization of the Property_representation class.

Value_range_property attributes:
- Property_determination (inherited from class Property_representation)
- Property.creation_date (inherited from class Property_representation)
- Upper_limit_value
- Lower_limit_value
- Unit

2.3.8.7 Text_property
The Text_property class is used to represent a property value described as a string value, eg "Green"

The Text_property class is a specialization of the Property_representation class.

Text_property attributes:
- Property_determination (inherited from class Property_representation)
- Property_creation_date (inherited from class Property_representation)
- Text

2.4 Unknown values
When no data can be provided, the following convention is recommended to be used:
- String '/NULL' indicates that data for a mandatory attribute is currently not known.
- '$' indicates that an optional attribute is not instantiated.

Note
This guidance has been derived from the "PDM Schema usage guide".

3 Model overview
The S3000L data model is organized into a set of Units of Functionalities (UoF), which splits the overall data model into a set of smaller data models. The purpose of this is to present small and coherent portions of the data model, and to gradually give the reader an understanding of the complete data model.

The figure below provides an overview of the main UoFs, and how they are interrelated:
4 Functional areas

4.1 UoF Project

4.1.1 Overall description

The first data elements defined during LSA are those required for defining the project (often referred to as the LSA program) and the product in focus (often referred to as the end item).

A project can be related to one or many contracts, where contracts can be organized into a chain of associated contracts, e.g., subcontracts.

Each contract is associated with at least one customer and one contractor. Each contract includes one or many contracted product variants, where a contracted product variant relates to a specific variant of a generic product.

A contracted product variant may also include information on the quantity of contracted product variant and/or the serial number range, as known by the customer.
4.1.2 Graphical representation

![Diagram of Project - class model](image)

4.1.3 UoF Project - New class and interface definitions

4.1.3.1 Project

The Project class identifies the overall LSA project (often also known as the LSA program).

Project attributes:
- Project_identifier (zero or one)
- Project_name

Project associations:
- A project must be associated with at least one contract.

4.1.3.2 Contract

The Contract class identifies contracts that are associated with the identified project.

Contract attributes:
- Contract_identifier (one or many)

Contract associations:
- Each contract can be associated with zero, one or many projects
− Each contract can refer to zero, one or many other contracts (via the associated_contract association)
− Each contract can be referred to from zero, one or many other contracts (via the associated_contract association)
− Each contract must be associated with at least one product variant
− Each contract must have at least one customer
− Each contract must have at least one contractor

4.1.3.3 Contract_relationship
The Contract_relationship class is an association class, which adds information to the association between two contracts.

Contract_relationship attributes:
− Contract_relationship_type

4.1.3.4 Organization
The Organization class identifies organizations that can be referred to from the LSA project.

Organization attributes:
− Organization_identifier (zero, one or many)
− Organization_name

An Organization class can be specialized into the following classes (also referred to as subclasses):
− Contractor
− Customer
− User

Each of these specializations has their own characteristics in addition to these that are generic for an Organization such as having an Organization_identifier and an Organization_name.

4.1.3.5 Contractor
The Contractor class is a specialization of class Organization, and identifies organizations that are contractors for one or many contracts.

Contractor attributes:
− Organization_identifier (inherited from class Organization)
− Organization_name (inherited from class Organization)

Contractor associations:
− A contractor can be associated with one or many contracts

4.1.3.6 Customer
The Customer class is a specialization of class Organization, and identifies organizations that are customers of one or many contracts.

Customer attributes:
− Organization_identifier (inherited from class Organization)
− Organization_name (inherited from class Organization)

Customer associations:
− A customer can be associated with one or many contracts
− A customer can have zero, one or many user organizations
4.1.3.7 User
The User class is a specialization of class Organization, and identifies organizations that are users within a specific customer organization.

User attributes:
- Organization_identifier (inherited from the class Organization)
- Organization_name (inherited from the class Organization)

User associations:
- A user must be associated with a specific customer
- A user can be associated with zero, one or many Contracted_product_variants, which the user is going to operate

4.1.3.8 Contracted_product_variant
The Contracted_product_variant class is an association class, which adds information to the association between a contract and the product variant that is contracted. A contracted product variant may be restricted to one or many specific blocks of serialized items, and/or to one or many specific user organizations.

Contracted_product_variant attributes:
- Quantity_of_contracted_product_variant (zero or one).

Contracted_product_variant associations:
- A Contracted_product_variant can be directed towards zero, one or many specific user organizations
- A Contracted_product_variant can be associated with zero, one or many Block_of_serialized_items

4.1.3.9 Block_of_serialized_items
The Block_of_serialized_items class defines a block of serialized items by serial numbers. The identification of a Block_of_serialized_items is always done from a Contracted_product_variant point of view, ie an instance of the Block_of_serialized_items class doesn’t have any meaning by itself but only in the context of a specific contracted product variant.

Block_of_serialized_items attributes:
- Serial_number_lower_bound
- Serial_number_upper_bound (zero or one)

Note
The value of the Serial_number_upper_bound attribute need not be specified. If the value for this attribute is not specified, the interval has no upper bound.

Block_of_serialized_items associations:
- A Block_of_serialized_items is always defined for a specific Contracted_product_variant.

4.1.3.10 Product
The Product class identifies a generic product, for which there must be at least one defined product variant.

Note
An example of generic product is F-15.

Product attributes:
- Product_identifier (one or many)
- Product_name (zero, one or many)

Note
An example of a type of Product_identifier is the GEIA-STD-0007 End Item Acronym Code.

Product associations:
- A Product must have at least one defined Product_variant

4.1.3.11 Product_variant
The Product_variant class identifies a specific variant of a defined generic product.

Note
A Product_variant is also known as a Model in S1000D and S2000M.

Note
Examples of product variants are: F15A and F15B

Product_variant attributes:
- Product_variant_identifier (one or many)
- Product_variant_name (zero, one or many)

Note
An example of a type of Product_variant_identifier is the GEIA-STD-0007 System/End Item Usable On Code.

Product_variant associations:
- Each Product_variant must be of a defined Product
- A Product_variant can be the end item that is identified in zero, one or many contracts

4.2 UoF Product Usage
4.2.1 Overall description
The Product Usage UoF defines the conditions under which the contracted product variant is to be operated and maintained. This includes eg maintenance levels and operating locations.

Note
The Maintenance_location and Operating_location classes are optional, and need not be defined within the LSA project.
4.2.2 Graphical representation

Fig 17 Product Usage - class model

4.2.3 UoF Product Usage - New class and interface definitions

4.2.3.1 Operator

The Operator interface must be implemented by those classes that can be used for defining the organization responsible for operating the contracted product variant. This model allows for both customers and/or users to be defined as Operators.

Classes that implement the Operator interface:

- Customer
- User

The Operator interface does not define any additional attributes.

Classes that implement the Operator interface must implement the following associations:

- An optional association with zero, one or many instances of Operating_location_type
- An optional association with zero, one or many instances of Operating_location
- An optional association with zero, one or many instances of Maintenance_level_type
- An optional association with zero, one or many instances of Maintenance_location

Note

Each Operating_location_type, Operating_location, Maintenance_level_type and Maintenance_location respectively, is uniquely defined for a specific Operator.

4.2.3.2 Maintenance_level_type

Maintenance activities may be structured in maintenance levels according to the customer or user maintenance policy, and to the customer organization of maintenance means.
Maintenance_level_type attributes:
- Maintenance_level_type_identifier (zero or one)
- Maintenance_level_type_name
- Maintenance_level_type_capability_description (zero or one)

Maintenance_level_type associations:
- Each Maintenance_level_type is uniquely defined for a specific Operator
- A Maintenance_level_type can be located together with zero, one or many Operating_location_types
- Each Maintenance_level_type can be associated with zero, one or many specific Maintenance_locations

4.2.3.3 Maintenance_location
The Maintenance_location class identifies actual locations where maintenance activities are to be carried out.

Maintenance_location attributes:
- Maintenance_location_identifier (zero or one)
- Maintenance_location_name
- Maintenance_location_description (zero or one)

Maintenance_location associations:
- Each Maintenance_location is uniquely defined for a specific Operator
- Each Maintenance_location belongs to a specific Maintenance_level_type

4.2.3.4 Operating_location_type
The Operating_location_type class defines types of locations where the contracted product variant will be operated.

Note
The Operating_location_type can be generic, i.e., it doesn't define any specific characteristics, but just enables the recording of the number of contracted product variants and its operating requirements as defined within the Contracted_product_variant_in_operating_location_type class.

Operating_location_type attributes:
- Operating_location_type_identifier (zero or one)
- Operating_location_type_name
- Operating_location_type_description (zero or one)
- Number_of_operating_locations (zero or one)

Operating_location_type associations:
- Each Operating_location_type is uniquely defined for a specific Operator
- An Operating_location_type must operate at least one Contracted_product_variant
- An Operating_location_type can be located together with zero, one or many Maintenance_level_types
- An Operating_location_type can be associated with zero, one or many specific Operating_locations

4.2.3.5 Operating_location
The Operating_location class defines the exact location where the contracted product variant will be operated.

Operating_location attributes:
− Operating_location_identifier (zero or one)
− Operating_location_name
− Operating_location_description (zero or one)

Operating_location associations:
− Each Operating_location is uniquely defined for a specific Operator
− A defined Operating_location must operate at least one Contracted_product_variant
− An Operating_location can be of a specific Operating_location_type

4.2.3.6 Contracted_product_variant_in_operating_location_type
The Contracted_product_variant_in_operating_location_type class is an association class, which adds information to the association between a contracted product variant and its operating location type.

Contracted_product_variant_in_operating_location_type attributes:
− Quantity_of_product_variant_in_operating_location_type (zero or one)
− Operating_requirement_in_operating_location_type (zero or one)

4.2.3.7 Contracted_product_variant_in_operating_location
The Contracted_product_variant_in_operating_location class is an association class, which adds information to the association between a contracted product variant and its operating location.

Contracted_product_variant_in_operating_location attributes:
− Quantity_of_product_variant_in_operating_location (zero or one)
− Operating_requirement_in_operating_location (zero or one)

4.2.4 UoF Product Usage - Additions to referenced class and interface definitions
4.2.4.1 Contracted_product_variant
The Contracted_product_variant class defined in UoF Project, must also implement the following additional associations:
− An association with one or many instances of Operating_location_type, where the Contracted_product_variant is to be operated
− An optional association with zero, one or many instances of Operating_location, where the Contracted_product_variant can be operated

4.2.4.2 Customer
The Customer class defined in UoF Project, must also implement the following additional interface:
− Operator

4.2.4.3 User
The user class defined in UoF Project, must also implement the following additional interface:
− Operator

4.3 UoF Breakdown Structure
4.3.1 Overall description
The Breakdown Structure UoF can be used to define one or many types of breakdowns for a specific Product. A breakdown can be eg functional, physical, system, zonal, or hybrid (ie a mixture of breakdown types).
Note
Existing breakdown structures are often hybrids consisting of both functional, physical and system elements.

Each breakdown may have one or many breakdown revisions (design iterations), where each breakdown revision references the breakdown element revisions that are included in the specific revision of the breakdown. These references are done using the Breakdown_element_usage_in_breakdown class. These references to the breakdown elements (revisions) are then organized hierarchically using the Breakdown_element_structure class.

Note
Having organized the breakdown element usages hierarchically by the use of Breakdown_element_structure, there needs to be one, and only one, instance of Breakdown_element_usage_in_breakdown that has no parent element. This Breakdown_element_usage_in_breakdown is often referred to as the root node.

This construct allows for a defined breakdown element (revision) to be part of many breakdowns and revisions thereof. A breakdown element (revision) can then be positioned differently in the respective breakdown and revision thereof. For example, a breakdown element (revision) can be used in both a functional and a physical breakdown.

Note
Each instance of Breakdown_element_usage_in_breakdown and Breakdown_element_structure is unique for a specific breakdown revision.
4.3.2 Graphical representation

4.3.3 UoF Breakdown Structure - New class and interface definitions

4.3.3.1 Breakdown

The breakdown class identifies a partitioning of a Product into a set of related elements so as to form explicit, parent-child relationships that comprise the breakdown.

Examples of breakdowns are:

- Functional
- Physical
- System
- Zonal
- Hybrid, i.e., a mixture of the above

Breakdown attributes:

- Breakdown_type.

Note

One example of a Breakdown_type is the GEIA-STD-0007 functional breakdown.

Breakdown associations:

- An association with the Product for which the breakdown is defined

Applicable to: All
4.3.3.2 Breakdown_revision
The Breakdown_revision class defines an explicit revision (design iteration) of a breakdown for a Product.

Note:
Breakdown_element_revisions used in a breakdown are related to a Breakdown_revision and not to the Breakdown.

Breakdown_revision attributes:
- Breakdown_revision_identifier
- Breakdown_revision_creation_date (zero or one)
- Breakdown_revision_status (zero or one)

Breakdown_revision associations:
- A Breakdown_revision is always associated with one and only one breakdown of a Product
- A Breakdown_revision is related to one or many Breakdown_element_revisions which are used in the breakdown revision, via the Breakdown_element_usage_in_breakdown class.

4.3.3.3 Breakdown_element_usage_in_breakdown
The Breakdown_element_usage_in_breakdown class defines a relationship between a revision of a breakdown element and a revision of a breakdown in which the revision of the breakdown element is being used.

Note
Each instance of Breakdown_element_usage_in_breakdown is uniquely defined for one combination of Breakdown_element_revision and Breakdown_revision. An instance of the Breakdown_element_usage_in_breakdown class doesn't have any meaning by itself but only in the context of a Breakdown_revision.

Breakdown_element_usage_in_breakdown associations:
- An association with the Breakdown_revision in which the associated Breakdown_element_revision is being used
- An association with the Breakdown_element_revision that is being used in the associated Breakdown_revision
- An optional parent-of association with zero, one or many Breakdown_element_revisions included in the Breakdown_revision, via the Breakdown_element_structure class
- An optional child-of association with zero or one Breakdown_element_revision included in the same Breakdown_revision, via the Breakdown_element_structure class

Note
An instance of Breakdown_element_usage_in_breakdown can be the parent of many instances of Breakdown_element_usage_in_breakdown (i.e., Breakdown_element_revisions) defined for the same Breakdown_revision.

Note
There is one instance of Breakdown_element_structure class per parent-child relationship, and there is no explicit ordering in between the parent-child relationships defined for a specific Breakdown_element_usage_in_breakdown (i.e., ordering of child elements for a specific parent element).

4.3.3.4 Breakdown_element_structure
The Breakdown_element_structure class is being used to hierarchically organize Breakdown_element_revisions within a specific Breakdown_revision.
Note
The breakdown structure is accomplished by organizing the usages of the Breakdown_element_revisions, ie by organizing the instances of Breakdown_element_usage_in_breakdown instead of the organizing the Breakdown_element_revisions themselves.

Note
The breakdown structure can also be derived from the values given as Breakdown_element_identifiers. The breakdown structure is then implicit, rather than explicit and do not require the usage of Breakdown_element_structure class. This approach can be used if the Breakdown_element_identifier follows the logic defined in eg GEIA-STD-0007 Logistics Control Number (LCN), A discussion on the advantages/disadvantages of the respective approach is given in Chap 3 LSA Business Process.

Breakdown_element_structure attributes:
- Quantity_of_child_elements (zero or one)
- Installation_description (zero or one)

Breakdown_element_structure associations:
- A parent association with an instance of Breakdown_element_usage_in_breakdown, representing the Breakdown_element_revision that is defined as the parent element in the associated Breakdown_revision
- A child association with an instance of Breakdown_element_usage_in_breakdown, representing the Breakdown_element_revision that is defined as the child element in the associated Breakdown_revision

4.3.3.5 Breakdown_element
The Breakdown_element class is used to represent eg a system, subsystem, function, zone, hardware item, software item, etc, that is part of one or many Breakdowns and revisions thereof.

Note
Breakdown_element is defined as an abstract class, ie this class will never be instantiated, but any of its subclasses will. Subclasses of Breakdown_element are:
- Hardware_element (defined in UoF Breakdown Element Realization below)
- Software_element (defined in UoF Breakdown Element Realization below)
- Zone_element (defined in UoF Breakdown Zone Element below)
- Aggregated_element (defined in UoF Breakdown Aggregated Element below)

Breakdown_element attributes:
- Breakdown_element_identifier (one or many)
- Breakdown_element_name (zero , one or many)

Breakdown_element associations:
- An association with the different Breakdown_element_revisions (design iterations) of the Breakdown_element
- A Breakdown_element can relate to one or many other Breakdown_elements via the Breakdown_element_relationship association
- A Breakdown_element can be related to from one or many other Breakdown_elements via the Breakdown_element_relationship association

4.3.3.6 Breakdown_element_revision
The Breakdown_element_revision class defines a revision (design iteration) of a breakdown element.
Note
Revisions of Breakdown_elements are often defined during product design.

Note
Breakdown_element_revision is defined as an abstract class, ie this class will never be instantiated but any of its subclasses will. Subclasses of Breakdown_element_revision are:

- Hardware_element_revision (defined in UoF Breakdown Element Realization below)
- Software_element_revision (defined in UoF Breakdown Element Realization below)
- Zone_element_revision (defined in UoF Breakdown Zone Element below)
- Aggregated_element_revision (defined in UoF Breakdown Aggregated Element below)

Breakdown_element_revision attributes:
- Breakdown_element_revision_identifier
- Breakdown_element_revision_creation_date (zero or one)
- Breakdown_element_revision_status (zero or one)
- Maintenance_significant_or_relevant_indicator

Breakdown_element_revision associations:
- An association with the Breakdown_element of which the Breakdown_element_revision is a revision
- Associations with one or many Breakdown_element_usage_in_breakdown, ie usages of the Breakdown_element_revision in one or many Breakdown_revisions of a Product

Note
A revision of a breakdown element can be used in one or many breakdowns via the Breakdown_revision and Breakdown_element_usage_in_breakdown. This means that the same Breakdown_element_revision may be re-used in between different types of breakdowns for a single Product, as well as be re-used in between Products.

4.3.3.7 Breakdown_element_relationship
The Breakdown_element_relationship class is an association class, which adds information to a defined relationship between two Breakdown_elements.

Note
Breakdown_element_relationship can be used to represent the functional/physical breakdown element relationship as defined in GEIA-STD-0007.

Note
Breakdown_element_relationship can be used to represent, eg the use of software in hardware.

Breakdown_element_relationship attributes:
- Breakdown_element_relationship_type

4.3.4 UoF Breakdown Structure - Additions to referenced class and interface definitions
4.3.4.1 Product
The Product class defined in UoF Project, must also implement the following additional association:
- An association with one or many Breakdowns

4.4 UoF Part
4.4.1 Overall description
The UoF Part is used to represent hardware items as well as software.
Part in S3000L represents an abstract class, and an instance of a part must be either a Hardware_part or a Software_part.

Parts can be used to realize breakdown elements in a breakdown of a product, as well as play the role of a resource in a maintenance task.

A part can have one or more alternate parts. The alternate part is interchangeable with the base part in any/all uses.

Parts can be organized into Bill-Of-Materials (BOM), ie parts can consist of other parts.

A part used in a Bill-Of-Material (BOM) can have one or many substitute parts. As opposed to the alternate part, which is context free, this substitute part is context dependent. The substitute part is only valid in the context of its usage as a component within the specified BOM.

Materials (often hazardous) are defined separately, and are then associated with the parts in which they are being used.
4.4.2 Graphical representation

**Part**

- Part_identifer: Id [1..*]
- Part_name: Id [1..*]
- Part_specification: Description [0..*]
- Part_maturity: Classification [0..*]
- Part_document: id: Identification [0..*]
- Part_document: classification: Classification [0..*]

**Hardware_part**

- Hardware_part: logistics_category: Class
- Hardware_part: Repeatability: Class [0..*]
- Hardware_part: maintainability: Class [0..*]
- Hardware_part: planned_disposal: Description [0..*]
- Hardware_part: planned_disposal: description: Description [0..*]

**Software_part**

- Software_part: type: Class
- Software_part: release: Id [0..*]
- Software_part: version: Id [0..*]
- Software_part: intended_use: Id [0..*]
- Software_part: intended_use: description: Description [0..*]
- Software_part: intended_use: classification: Classification [0..*]
- Software_part: intended_use: description: Description [0..*]

**Hardware_part: operational authorized life**

- Hardware_part: operational authorized life: Id [0..*]

**Material**

- Material: id: Id [0..*]
- Material: name: Id [0..*]
- Material: description: Description [0..*]
- Material: substance_usage: category: Class
- Material: substance_usage: description: Description [0..*]
- Material: characteristic: Class [0..*]
- Material: characteristic: description: Description [0..*]

**Fig 19 Part - class model**

4.4.3 UoF Part - New class and interface definitions

4.4.3.1 Part

Part is a discrete object that may come into existence as a consequence of a manufacturing process (Source: ISO 10303:239 Product Life Cycle Support).

A part can be at any indenture level in a part structure, ie a part can be an assembly as well as an equipment, component or software package.
Note
Part is an abstract class, ie an instantiation of Part needs to be either a Hardware_part or a Software_part.

Part attributes:
- Part_identifier (one or many)
- Part_name (one or many)
- Part_special_handling (zero or one)
- Part_maturity (zero, one or many)
- Part_obsolescence_risk_assessment (zero, one or many)
- Part_demilitarization_class (zero or one)

Note
Examples of different Part_identifiers are; Original Equipment Manufacturer (OEM) part numbers, Supplier part numbers etc.

Note
A part can have more than one Part_maturity classification. This will most likely reflect changes in part maturity over time. If the Date attribute included in the Classification data type is not enough to determine the validity of the respective Part_maturity value, then Applicability_statement (refer to UoF Applicability statement) can be used to further determine the applicability of the respective value.

Note
A part can have more than one Part_obsolescence_risk_assessment description. This will most likely reflect changes in part obsolescence risk assessment over time. If the Date attribute included in the Classification data type is not enough to determine the validity of the respective Part_obsolescence_risk_assessment value, then Applicability_statement (refer to UoF Applicability statement) can be used to further determine the applicability of the respective value.

Part associations:
- A part can be associated with zero, one or many other parts, which the part under consideration consists of (via the parent association with the Parts_list_entry class)
- A part can be associated with zero, one or many other parts, which the part under consideration is part of (via the child association with the Parts_list_entry class)
- A part can refer to zero, one or many alternate parts (via the base association with the Alternate_part_relationship class)
- A part can be an alternate part for one or many other parts (defined via the alternate association with the Alternate_part_relationship class)

4.4.3.2 Alternate_part_relationship
The Alternate_part_relationship class is used to define alternate parts. An alternate part is interchangeable with the base part in all its usages. The alternate part is a context independent alternate that is form, fit, and function equivalent in any use (Source: ISO 10303:239 Product Life Cycle Support).

Note
Alternate_part_relationship defines one alternate Part at the time, and applies both to Hardware_parts and Software_parts.

Note
Each instance of Alternate_part_relationship may have an associated Applicability statement (refer to UoF Applicability Statement).

Alternate_part_relationship associations:
− An association with the base (primary) Part for which an alternate is specified
− An association with the Part that may replace the base Part in all its usages.

4.4.3.3 Parts_list_entry

The Parts_list_entry class is being used to hierarchically organize Bill-of-Materiels (BOM) for a Part.

Note
Parts_list_entry defines one “row” at the time in a Bill-of-Material, ie there is one instance of Parts_list_entry per contained part. There is no explicit class in the data model that represents a complete Bill-of-Material.

Note
The Parts_list_entry applies both to Hardware_parts and Software_parts, and a Bill-of-Material may consist of both Hardware_parts and Software_parts.

Parts_list_entry attributes:
− Parts_list_entry_identifier (zero, one or many)
− Quantity_of_child_elements
− Physical_replaceability
− Replaceability_strategy (zero, one or many)

Note
Physical_replaceability and Replaceability_strategy in the context of a Parts_list_entry are used to define if the child part can be replaced in its parent part, and if so at what maintenance level.

Parts_list_entry associations:
− An association with an instance of Part that is the parent part in the Bill-of-Material relationship
− An association with an instance of Part that is the child part in the Bill-of-Material relationship
− An association with zero, one or many Parts_list Entries that specifies substitute Parts for a specific Parts_list_entry (defined using the base_parts_list_entry association with the Substitute_part_relationship class)
− An association with zero or one base Parts_list_entry, for which the Parts_list_entry under consideration may be used in place of (defined using the substitute_part_list_entry association with the Substitute_part_relationship class)

Parts_list_entry special recommendations:
− A Part included in another Part can have multiple Replaceability_strategies (ie values of the attribute Replaceability_strategy). Each Replaceability_strategy value (ie instance of data type Class) must have an associated Applicability_statement, as defined in the UoF Applicability Statement, in order to define when the respective Replaceability_strategy is applicable. The applicability statement will in most cases include Operator as one of the parameters to be evaluated.

4.4.3.4 Substitute_part_relationship

The Substitute_part_relationship class is used to define substitute Parts. As opposed to the alternate part, which is context free, substitute part is context dependent, ie the Substitute_part_relationship is a relationship that indicates that one Parts_list_entry instance can be substituted by another instance of Parts_list_entry, where both instances of Parts_list_entry refer to the same parent Part.  (Source: ISO 10303:239 Product Life Cycle Support).
Note

Substitute_part_relationship defines one substitute Part at the time, and applies both to Hardware_parts and Software_parts.

Substitute_part_relationship attributes:

- Parts_list_entry_substitute_identifier

Substitute_part_relationship associations:

- An association with the base Parts_list_entry for which a substitute is specified
- An association with the substitute Parts_list_entry which may replace the base Parts_list_entry.

4.4.3.5 Hardware_part

The Hardware_part class is a specialization of class Part, and represents those parts that are realized as physical items (hardware).

Hardware_part attributes:

- Part_identifier (inherited from class Part)
- Part_name (inherited from class Part)
- Part_special_handling (inherited from class Part)
- Part_maturity (inherited from class Part)
- Part_obsolescence_risk_assessment (inherited from class Part)
- Part_demilitarization_class (inherited from class Part)
- Hardware_part_logistics_category
- Repairability
- Repairability_strategy (zero, one or many)
- Hardware_part_maintenance_start (zero or one)
- Hardware_part_material_hazardous_class (zero, one or many)
- Hardware_part_waste_products_in_use_disposal_description (zero, one or many)
- Hardware_part_waste_products_planned_disposal_description (zero, one or many)
- Hardware_part_environmental_aspect_class_in_use (zero, one or many)
- Hardware_part_environmental_aspect_class_planned_disposal (zero, one or many)
- Hardware_part_scrap_rate (zero or one)

Note

A part can have more than one
Hardware_part_waste_products_in_use_disposal_description and/or
Hardware_part_waste_products_planned_disposal_description. This will most likely reflect changes in part waste products disposal descriptions over time.

Note

A part can have more than one Hardware_part_environmental_aspect_class_in_use and/or
Hardware_part_environmental_aspect_class_planned_disposal classification. This will most likely reflect changes in hardware part environmental aspects over time.

Hardware_part associations:

- A Hardware_part can be associated with zero, one or many other parts (Hardware_parts and/or Software_parts), which the Hardware_part under consideration consists of (inherited from class Part)
- A Hardware_part can be associated with zero, one or many other parts (Hardware_parts and/or Software_parts), which the Hardware_part under consideration is part of (inherited from class Part)
- A Hardware_part can refer to zero, one or many alternate Hardware_parts (inherited from Part class)
- A Hardware_part can be an alternate Hardware_part for one or many other Hardware_parts (inherited from Part class)
- A Hardware_part can have zero, one or many defined operational authorized lifes, as defined by the Hardware_part_operational Authorized_life class
- A Hardware_part can have zero, one or many associated Materials, included in the Hardware_part

**Hardware_part special recommendations:**
- A Hardware_part can have multiple Repairability_strategies (ie values for Repairability_strategy). Each Repairability_strategy value (ie instance of the Class data type) must have an associated Applicability_statement, as defined in the UoF Applicability Statement, in order to define when respective Repairability_strategy is applicable. The applicability statement will in most cases include Operator as one of the parameters to be evaluated.

### 4.4.3.6 Hardware_part_operationalAuthorized_life

Operational authorized life consists of the value pair for operational authorized life and the authorizing organization.

**Note**

The same authorizing organization can be related to one or more operational authorized life values for a Hardware_part since the Hardware_part_operationalAuthorized_life attribute can have more than one value (ie instances of the Prp class).

**Note**

An instance of class Hardware_part_operationalAuthorized_life has no meaning by itself, but only in the context of a specific Hardware_part.

**Hardware_part_operationalAuthorized_life attributes:**
- Hardware_part_operationalAuthorized_life (one or many)

**Hardware_part_operationalAuthorized_life associations:**
- An instance of Hardware_part_operationalAuthorized_life is always associated with one authorizing Organization
- An instance of Hardware_part_operationalAuthorized_life is always associated with a specific Hardware_part

**Hardware_part_operationalAuthorized_life special recommendations:**
- Representations of the value for the Hardware_part_operationalAuthorized_life attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to "maximum". There can also be cases where use of Value_with_tolerances_property or Value_range_property is valid.
- Where a Hardware_part has different operational authorized lives depending on, for example, operational environments, these can be represented as a set of values of the Hardware_part_operationalAuthorized_life attribute, but be distinguished by the assignment of an Applicability_statement to the respective instance of class Prp that represents the individual value (also refer to Para 4.21, UoF Applicability Statement).

### 4.4.3.7 Material

The Material class deals with substances (mainly hazardous) that are used in one or many Hardware_parts.

**Material attributes:**
- Material_identifier (one or many)
- Material_name (zero or one)
− Material_description (zero or one)
− Material_substance_usage_category
− Material_risk_description (one or many)
− Material_risk_factor (zero or one)
− Material_characteristics_recording_date

Material associations:
− A Material can have zero, one or many associations with Hardware_parts in which the Material is being used.

4.4.3.8 Hardware_part_material_usage
The Hardware_part_material_usage class is an association class, which adds information to a defined relationship between a Material and the Hardware_part in which the Material is being used. It can also record the quantity of the Material being used in the Hardware_part, and the justification for its usage.

Hardware_part_material_usage attributes:
− Quantity_of_material_included_in_hardware_part
− Material_usage_justification_description (zero or one)

Note
There is one instance of Hardware_part_material_usage per relevant combination of Material and Hardware_part.

4.4.3.9 Software_part
Software_part is a specialization of class Part, and represents those parts that are realized as executable software or as data files (eg maps).

Software_part attributes:
− Part_identifier (inherited from class Part)
− Part_name (inherited from class Part)
− Part_special_handling (inherited from class Part)
− Part_maturity (inherited from class Part)
− Part_obsolescence_risk_assessment (inherited from class Part)
− Part_demilitarization_class (inherited from class Part)
− Software_type
− Software_release
− Actual_software_size (zero or one)
− Actual_number_of_lines_of_code (zero or one)
− Actual_software_complexity_code (zero or one)
− Software_technology (zero or one)

Software_part associations:
− A Software_part can be associated with zero, one or many other parts (Hardware_parts and/or Software_parts), which the Software_part under consideration consists of (inherited from class Part)
− A Software_part can be associated with zero, one or many other parts (Hardware_parts and/or Software_parts), which the Software_part under consideration is part of (inherited from class Part)
− A Software_part can refer to zero, one or many alternate Software_parts (inherited from class Part)
− A Software_part can be an alternate Software_part for one or many other Software_parts (inherited from class Part)
4.4.4 UoF Part - Additions to referenced class and interface definitions

4.4.4.1 Organization

The Organization class defined in UoF Project, must also implement the following additional association:

- An optional association with zero, one or many instances of Hardware_part_operational_authorised_life, in the role of authorizing_organization

4.5 UoF Breakdown Element Realization

4.5.1 Overall description

The Breakdown Element Realization UoF is used to:

- Define hardware and software breakdown elements along with their realizations in terms of hardware and software parts, respectively
- Distinguish hardware and software breakdown elements from other types of breakdown elements such as systems, zones

Note

The Hardware_element and Software_element classes are both specializations of the class Breakdown_element which means that, wherever the class Breakdown_element is being used in the data model, Hardware_element or Software_element can be used instead (the rule of substitutability).

The Hardware_element_revision and Software_element_revision classes are both specializations of the class Breakdown_element_revision which means that, wherever the class Breakdown_element_revision is being used in the data model, Hardware_element_revision or Software_element_revision can be used instead (the rule of substitutability).
4.5.2 Graphical representation

Fig 20 Breakdown Element Realization - class model

4.5.3 UoF Breakdown Element Realization - New class and interface definitions

4.5.3.1 Hardware_element

The Hardware_element class is a specialization of Breakdown_element, and is used to represent physical breakdown elements that are to be realized as hardware parts (often referred to as equipments).

Hardware_element attributes:
- Breakdown_element_identifier (inherited from class Breakdown_element)
- Breakdown_element_name (inherited from class Breakdown_element)
- Hardware_element_type (zero or one)

Hardware_element associations:
- An association with the different Hardware_element_revisions (design iterations) of the Hardware_element
− A Hardware_element can relate to one or many other Breakdown_elements via the Breakdown_element_relationship (inherited from class Breakdown_element)
− A Hardware_element can be related to from one or many other Breakdown_elements via the Breakdown_element_relationship (inherited from class Breakdown_element)

4.5.3.2 Hardware_element_revision
The Hardware_element_revision class is a specialization of Breakdown_element_revision, and is used to represent revisions of physical breakdown elements that are to be realized as Hardware_parts.

Hardware_element_revision attributes:
− Breakdown_element_revision_identifier (inherited from class Breakdown_element_revision)
− Breakdown_element_revision_creation_date (inherited from class Breakdown_element_revision)
− Breakdown_element_revision_status (inherited from class Breakdown_element_revision)
− Maintenance_significant_or_relevant_indicator (inherited from class Breakdown_element_revision)
− Structural_indicator
− Functional_replaceability
− Replaceability_strategy (zero, one or many)
− Repairability
− Repairability_strategy (zero, one or many)

Hardware_element_revision associations:
− An association with the Hardware_element of which the Hardware_element_revision is a revision
− Associations with one or many Breakdown_element_usage_in_breakdown, ie usages of the Hardware_element_revision in one or many Breakdown_revisions of a Product (inherited from class Breakdown_element_revision)
− An optional association with zero, one or many Hardware_parts that can be used to realize the defined Hardware_element_revision, via the Hardware_element_realization class

Hardware_element_revision special recommendations:
− A Hardware_element_revision can have multiple Replaceability_strategies (ie values of the attribute Replaceability_strategy). Each Replaceability_strategy value (ie instance of Class) must have an associated Applicability_statement, as defined in the UoF Applicability Statement, in order to define when the respective Replaceability_strategy is applicable. The applicability statement will in most cases include Operator as one of the parameters to be evaluated.
− A Hardware_element_revision can have multiple Repairability_strategies (ie values for Repairability_strategy). Each Repairability_strategy value (ie instance of Class) must have an associated Applicability_statement, as defined in UoF Applicability Statement, in order to define when respective Repairability_strategy is applicable. The applicability statement will in most cases include Operator as one of the parameters to be evaluated.

4.5.3.3 Hardware_element_realization
The Hardware_element_realization class defines an association between an instance of Hardware_element_revision and its Hardware_part realization.

Hardware_element_realization associations:
− An association with the instance of Hardware_element_revision that has a Hardware_part realization
− An association with the instance of Hardware_part that is the realization of the Hardware_element_revision
Note

There is one instance of Hardware_element_realization per relevant combination of Hardware_element_revision and Hardware_part.

Hardware_element_realization special recommendations:

- Where an instance of Hardware_element_realization is dependent upon Product_variant or Product_variant_realization, it must be distinguished by the assignment of a Product_variant_realization_applicability to the instance of Hardware_element_realization (refer to Para 4.8, UoF Product variant applicability).

4.5.3.4 Software_element

The Software_element class is a specialization of Breakdown_element, and is used to represent breakdown elements that are to be realized in terms of software.

Software_element attributes:

- Breakdown_element_identifier (inherited from class Breakdown_element)
- Breakdown_element_name (inherited from class Breakdown_element)
- Software_element_type (zero or one)

Software_element associations:

- An association with the different Software_element_revisions (design iterations) of the Software_element
- A Software_element can relate to one or many other Breakdown_elements via the Breakdown_element_relationship (inherited from class Breakdown_element)
- A Software_element can be related to from one or many other Breakdown_elements via the Breakdown_element_relationship (inherited from class Breakdown_element)

4.5.3.5 Software_element_revision

The Software_element_revision class is a specialization of Breakdown_element_revision, and is used to represent revisions (design iterations) of breakdown elements that are to be realized as Software_parts.

Software_element_revision attributes:

- Breakdown_element_revision_identifier (inherited from class Breakdown_element_revision)
- Breakdown_element_revision_creation_date (inherited from class Breakdown_element_revision)
- Breakdown_element_revision_status (inherited from class Breakdown_element_revision)
- Maintenance_significant_or_relevant_indicator (inherited from class Breakdown_element_revision)
- Estimated_software_size (zero or one)
- Estimated_number_of_lines_of_code (zero or one)
- Estimated_software_complexity_code (zero or one)
- Estimated_software_modification_frequency (zero or one)

Software_element_revision associations:

- An association with the Software_element of which the Software_element_revision is a revision
- Associations with one or many Breakdown_element_usage_in_breakdown, ie usages of the Software_element_revision in one or many Breakdown_revisions of a Product (inherited from class Breakdown_element_revision)
- An optional association with zero, one or many Software_parts that can be used to realize the defined Software_element_revision, via the Software_element_realization class
4.5.3.6 **Software_element_realization**

The *Software_element_realization* class defines an association between an instance of *Software_element_revision* and its *Software_part* realization.

**Software_element_realization** associations:

- An association with the instance of *Software_element_revision* that has a *Software_part* realization
- An association with the instance of *Software_part* that is the realization of the *Software_element_revision*

**Note**

There is one instance of *Software_element_realization* per relevant combination of *Software_element_revision* and *Software_part*.

**Software_element_realization** special recommendations:

- Where an instance of *Software_element_realization* is dependent upon *Product_variant* or *Product_variant_realization*, it must be distinguished by the assignment of a *Product_variant_realization_applicability* to the instance of *Software_element_realization* (refer to Para 4.8, *UoF Product Variant Applicability*).

4.5.4 **UoF Breakdown Element Realization - Additions to referenced class and interface definitions**

4.5.4.1 **Hardware_part**

The *Hardware_part* class defined in *UoF Part*, must also implement the following additional association:

- An optional association with zero, one or many instances of *Hardware_element_realization*

4.5.4.2 **Software_part**

The *Software_part* class defined in *UoF Part*, must also implement the following additional association:

- An optional association with zero, one or many instances of *Software_element_realization*

4.6 **UoF Breakdown Zone Element**

4.6.1 **Overall description**

The Breakdown Zone Element UoF is used to distinguish between zonal breakdown elements and other types of breakdown elements (eg hardware, software).

**Note**

Zone elements can be used to define eg zones within a product, as well as generic work areas, eg a mechanical workshop.
4.6.3 UoF Breakdown Zone Element - New class and interface definitions

4.6.3.1 Zone_element

The Zone_element class is a specialization of Breakdown_element, and is used to represent breakdown elements that are defined in terms of zones within a product, or work areas such as a mechanical workshop.

**Note**

Wherever the Breakdown_element class is used in the data model, Zone_element can be used instead (the rule of substitutability).

Zone_element attributes:

- Breakdown_element_identifier (inherited from class Breakdown_element)
- Breakdown_element_name (inherited from class Breakdown_element)
- Zone_element_type (zero or one)

Zone_element associations:

- An association with the different Zone_element_revisions (design iterations) of the Zone_element
- A Zone_element can relate to one or many other Breakdown_elements via the Breakdown_element_relationship (inherited from class Breakdown_element)
- A Zone_element can be related to from one or many other Breakdown_elements via the Breakdown_element_relationship (inherited from class Breakdown_element)

4.6.3.2 Zone_element_revision

The Zone_element_revision class is a specialization of Breakdown_element_revision, and is used to represent revisions of zone elements.
Note
Wherever the Breakdown_element_revision class is used in the data model, Zone_element_revision can be used instead (the rule of substitutability). For example, an instance of Zone_element_revision can be used in one or many types of breakdowns and revisions thereof.

Zone_element_revision attributes:
- Breakdown_element_revision_identifier (inherited from class Breakdown_element_revision)
- Breakdown_element_revision_creation_date (inherited from class Breakdown_element_revision)
- Breakdown_element_revision_status (inherited from class Breakdown_element_revision)
- Maintenance_significant_or_relevant_indicator (inherited from class Breakdown_element_revision)
- Zone_element Description (zero or one)

Zone_element_revision associations:
- An association with the Zone_element of which the Zone_element_revision is a revision
- Associations with one or many Breakdown_element_usage_in_breakdown, ie usages of the Zone_element_revision in one or many Breakdown_revisions of a Product (inherited from class Breakdown_element_revision).
- Associations with zero, one or many instances of Hardware_element_revision that are located in the zone, via the Hardware_element_in_zone_relationship class.

4.6.3.3 Hardware_element_in_zone_relationship
The Hardware_element_in_zone_relationship class defines an association between an instance of Hardware_element_revision and its zonal location.

Hardware_element_in_zone_relationship associations:
- An association with the instance of Hardware_element_revision that has a zonal location
- An association with the instance of Zone_element_revision in which the hardware element is located

Note
There is one instance of Hardware_element_in_zone_relationship per relevant combination of Hardware_element_revision and Zone_element_revision.

Hardware_element_in_zone_relationship special recommendations:
- Where a Hardware_element_in_zone_relationship is dependent upon a Product_variant, it must be distinguished by assigning a Product_variant_applicability to the Hardware_element_in_zone_relationship (refer to Para 4.8, UoF Product Variant Applicability).

4.6.4 UoF Breakdown Zone Element - Additions to referenced class and interface definitions
4.6.4.1 Hardware_element_revision
The Hardware_element_revision class defined in UoF Breakdown Element Realization, must also implement the following additional association:
- An optional association with zero, one or many Hardware_element_in_zone_relationship

4.7 UoF Breakdown Aggregated Element
4.7.1 Overall description
The Breakdown Aggregated Element UoF is used to distinguish between aggregated (or abstract) breakdown elements (systems, functions, slots etc) and other types of breakdown elements (eg hardware, software).
Note
Aggregated breakdown elements of type ‘System’ and ‘Function’ can have a set of child breakdown elements (refer to Para 4.3 UoF Breakdown Structure). These child elements are to be seen as a whole-parts relationship, ie all children must exist in order for the parent system or function to be complete (exception if there is any defined Product_variant_applicability statement as defined in Para 4.8 UoF Product Variant Applicability). However aggregated breakdown elements of type ‘Slot’ indicates that all child elements are alternative breakdown elements which can be used in the defined slot location.

4.7.2 Graphical representation

Fig 22  Breakdown Aggregated Element - class model

4.7.3 UoF Breakdown Aggregated Element - New class and interface definitions

4.7.3.1 Aggregated_element
The Aggregated_element class is a specialization of Breakdown_element, and is used for representing abstract breakdown elements, eg a system, function or a slot.

Note
Wherever the Breakdown_element class is used in the data model, Aggregated_element can be used instead (the rule of substitutability).

Aggregated_element attributes:
- Breakdown_element_identifier (inherited from class Breakdown_element)
- Breakdown_element_name (inherited from class Breakdown_element)
- Aggregated_element_type (zero or one)

Aggregated_element associations:
- An association with the different Aggregated_element_revisions (design iterations) of the Aggregated_element
- An Aggregated_element can relate to one or many other Breakdown_elements via the Breakdown_element_relationship (inherited from class Breakdown_element)
- An Aggregated_element can be related to from one or many other Breakdown_elements via the Breakdown_element_relationship (inherited from class Breakdown_element)

4.7.3.2 Aggregated_element_revision
The Aggregated_element_revision class is a specialization of Breakdown_element_revision, and is used to represent revisions of Aggregated_element.

Note
Wherever Breakdown_element_revision is used in the data model, Aggregated_element_revision can be used instead (the rule of substitutability). Eg an instance of Aggregated_element_revision can be used in one or many types of breakdowns (and revisions thereof).
Aggregated_element_revision attributes:
- Breakdown_element_revision_identifier (inherited from class Breakdown_element_revision)
- Breakdown_element_revision_creation_date (inherited from class Breakdown_element_revision)
- Breakdown_element_revision_status (inherited from class Breakdown_element_revision)
- Maintenance_significant_or_relevant_indicator (inherited from class Breakdown_element_revision)
- Aggregated_element_description (zero or one)

Aggregated_element_revision associations:
- An association with the Aggregated_element of which the Aggregated_element_revision is a revision
- Associations with one or many Breakdown_element_usage_in_breakdown, ie usages of the Aggregated_element_revision in one or many Breakdown_revisions of a Product (inherited from class Breakdown_element_revision)

4.8 UoF Product Variant Applicability

4.8.1 Overall description
The Product Variant Applicability UoF enables the definition of product models which are available or delivered to the customer in different configurations or variations. A variation or configuration of a product is defined by applying restrictions to the generic product breakdown and its breakdown elements realizations, respectively.

Note
Product data standards such as ISO 10303, often use the term effectivity to describe product design related restrictions, where the restriction defines the planned usage of components in the context of a particular product configuration.

The Product Variant Applicability UoF enables specification of valid use of:
- Breakdown elements in the context of a Breakdown_revision
- Parts in the context of a particular Product_variant_realization (also often referred to as a Product configuration)

This UoF defines two types of product design related applicability’s:

- Product_variant_applicability
  - Constrains the usage of a Breakdown_element_revision within a Breakdown_revision to a specific Product_variant (also known in GEIA-STD-0007 as System/End Item Usable On Code).
  - Constrains the zonal location (Zone_element_revision) for a Hardware_element_revision to a specific Product_variant.

- Product_variant_realization_applicability
  - Constrains a Hardware_element_realization, Software_element_realization and/or Parts_list_entry to a specific realization (configuration) of a Breakdown_revision.

Note
Information regarding Product_variant_applicability’s, and Product_variant_realization_applicability’s, is often received from the design department as part of the product definition. This information often originates from of some kind of PDM-system.

Note
The use of Product_variant_applicability’s, and Product_variant_realization_applicability’s, enable reuse of product breakdowns between different Product_variants and realizations thereof.
Note
If Product_variant_applicability's, and Product_variant_realization_applicability's are used to restrict the applicability of elements within breakdown, then it needs to be explicitly defined for each Breakdown_revision.

Product_variant_applicability's and Product_variant_realization_applicability's helps the LSA analyst to filter out the Breakdown_elements and/or Parts that are within scope of a specific Product_variant, and realization thereof.

Product_variant_applicability and Product_variant_realization_applicability are used to define restrictions for a given product design (product definition), whereas Applicability_statement (refer to Para 4.21, UoF Applicability Statement) is used to define usage related restrictions.

4.8.2 Graphical representation

Fig 23  Product Variant Applicability - class model

4.8.3 UoF Product Variant Applicability - New class and interface definitions

4.8.3.1 Product_variant_applicability

The Product_variant_applicability class defines that the usage of a Breakdown_element_revision, or a zonal location for a Hardware_element_revision, (applicability_target) is limited to a specific Product_variant.
Note
Product_variant_applicability can be used for representing GEIA-STD-0007 System/End Item Usable On Codes.

Note
Instances of Breakdown_element_usage_in_breakdown and Hardware_element_in_zone_relationship without any associated Product_variant_applicability, means that the respective usage and zonal location of the associated Breakdown_element_revision, is valid for all Product_variants.

Note
An instances of Breakdown_element_usage_in_breakdown and Hardware_element_in_zone_relationship that has an associated Product_variant_applicability, needs one instance of Product_variant_applicability per Product_variant for which the respective instance of Breakdown_element_usage_in_breakdown or Hardware_element_in_zone_relationship is applicable.

Product_variant_applicability associations:
- An association with the Product_variant for which the applicability_target is valid.
- An association with the applicability_target (ie an instance of any class that implements the Product_variant_applicability_interface), which is valid in the resolved Product_variant.

Note
There is one instance of Product_variant_applicability per relevant combination of Product_variant and applicability_target (ie instance of either a Breakdown_element_usage_in_breakdown or Hardware_element_in_zone_relationship).

4.8.3.2 Product_variant_applicability_interface
The Product_variant_applicability_interface is implemented by classes that can be the target for an instance of Product_variant_applicability.

Classes that implement the Product_variant_applicability_interface interface:
- Breakdown_element_usage_in_breakdown
- Hardware_element_in_zone_relationship

Classes that implement the Product_variant_applicability_interface must also implement the following associations:
- An optional association with zero, one or many instances of Product_variant_applicability.

Note
An instance of the class that is the applicability_target can have more than one associated Product_variant_applicability, ie the instance being the applicability_target is applicable within more than one Product_variant.

4.8.3.3 Product_variant_realization
Product_variant_realization defines an approved realization of a Breakdown_revision, and controls which parts are allowed to realize the respective Hardware_element_revision, Software_element_revision, and Parts_list_entry which are part of the defined Breakdown_revision, for a particular Product_variant. This is managed through the usage of the Product_variant_realization_applicability class.

Note
Product data standards such as ISO 10303 often use the term product configuration to describe the planned usage of components in the context of a particular product configuration.
Product_variant_realization attributes:
- Product_variant_realization_identifier
- Product_variant_realization_revision_identifier
- Product_variant_realization_revision_creation_date (zero or one)

Product_variant_realization associations:
- An association with the Breakdown_revision for which the Product_variant_realization is defined.
- Associations with the Hardware_element_realizations, Software_element_realizations and/or Parts_list_entries that is applicable for the Product_variant_realization under consideration (via Product_variant_realization_applicability).

4.8.3.4 Product_variant_realization_applicability
The Product_variant_realization_applicability class defines an association between the resolved Product_variant_realization and the elements (applicability_targets) which are applicable within the Product_variant_realization under consideration.

Product_variant_realization_applicability associations:
- An association with the Product_variant_realization for which the applicability_target is valid.
- An association with the applicability_target (ie an instance of any class that implements the Product_variant_realization_applicability_interface, ie an instance of either Hardware_element_realization, Software_element_realization or Parts_list_entry) which is valid in the resolved Product_variant_realization.

Note
There is one instance of Product_variant_realization_applicability per relevant combination of Product_variant_realization and applicability_target.

Note
An instance of Hardware_element_realization, Software_element_realization or Parts_list_entry that does not have an associated Product_variant_realization_applicability means that the instance is valid for all Product_variants and realizations thereof.

Note
An instances of Hardware_element_realization, Software_element_realization or Parts_list_entry that has an assigned Product_variant_realization_applicability, need an instance of Product_variant_realization_applicability_interface per Product_variant_realization for which the Hardware_element_realization, Software_element_realization or Parts_list_entry is applicable.

4.8.3.5 Product_variant_realization_applicability_interface
The Product_variant_realization_applicability_interface is implemented by classes that can be the target for a Product_variant_realization_applicability.

Classes that implement the Product_variant_realization_applicability_interface interface:
- Hardware_element_realization
- Software_element_realization
- Parts_list_entry

Classes that implement the Product_variant_realization_applicability_interface must also implement the following association:
- An optional association with zero, one or many instances of Product_variant_realization_applicability
Note

An instance of the class that is the applicability_target can have more than one associated
Product_variant_realization_applicability, ie the instance being the applicability_target is
applicable within more than one Product_variant_realization.

4.8.4 UoF Product Variant Applicability - Additions to referenced class and interface
definitions

4.8.4.1 Product_variant
The Product_variant class defined in UoF Project, must also implement the following additional
association:
− An optional association with zero, one or many instances of Product_variant_applicability

4.8.4.2 Breakdown_revision
The Breakdown_revision class defined in UoF Breakdown Structure must also implement the
following additional association:
− An optional association with zero, one or many instances of Product_variant_realization

4.8.4.3 Breakdown_element_usage_in_breakdown
The Breakdown_element_usage_in_breakdown class defined in UoF Breakdown Structure, must also implement the following additional interface:
− Product_variant_applicability_interface

4.8.4.4 Hardware_element_in_zone_relationship
The Hardware_element_in_zone_relationship class defined in UoF Breakdown Zone Element, must also implement the following additional interface:
− Product_variant_applicability_interface

4.8.4.5 Hardware_element_realization
The Hardware_element_realization class defined in UoF Breakdown Element Realization, must also implement the following additional interface:
− Product_variant_realization_applicability_interface

4.8.4.6 Software_element_realization
The Software_element_realization class defined in UoF Breakdown Element Realization, must also implement the following additional interface:
− Product_variant_realization_applicability_interface

4.8.4.7 Parts_list_entry
The Parts_list_entry class defined in UoF Part, must also implement the following additional
interface:
− Product_variant_realization_applicability_interface

4.9 UoF LSA Candidate

4.9.1 Overall description
The LSA Candidate UoF supports the selection of LSA candidates and the definition of the key
performance indicators per selected LSA candidate.

Note
The recommendation is to select the LSA Candidates from a top-down perspective, ie
starting with the root Breakdown Element node for a given Product_variant and then
traverse through the Breakdown Structure and its part realizations (including Parts_list_entries).

**Note**
Key performance indicators define the requirements for the project, and are often defined during the early stages of the LSA project (program).

**Note**
The process of distributing a key performance indicator between sub elements using eg fractions is not explicitly supported in the data model. However, the data model supports the recording of the resulting values against the respective sub element. Key performance parameters that are the result of a distribution process, eg by fraction, are recommended to set the value of the Key_performance_indicator_type attribute to “Distributed”.
4.9.2 Graphical representation

ICN-B6865-S3000L0223-002-01

Fig 24 UoF LSA Candidate - class model

4.9.3 UoF LSA Candidate - New class and interface definitions

4.9.3.1 LSA_candidate

The LSA_candidate interface is implemented by classes that can be selected as LSA Candidates.

Classes that implement the LSA_candidate interface:
- Breakdown_element_revision
- Part
Since the LSA_candidate interface is implemented by Breakdown_element_revision and Part respectively, it means that instances of the following classes can be selected as LSA Candidates: Hardware_element_revision, Software_element_revision, Aggregated_element_revision, Zone_element_revision, Hardware_part and Software_part.

Classes that implement the LSA_candidate interface must implement the following attributes:

- LSA_candidate_indicator
- LSA_candidate_rationale
- LSA_candidate_maintenance_concept (zero or one)
- LSA_candidate_maintenance_solution (zero or one)

Classes that implement the LSA_candidate interface must implement the following association:

- An optional association with zero, one or many instances of Key_performance_indicator

The associated Key_performance_indicators may be of any its specializations (eg Mean_time_between_failure, Down_time etc), as well as any number of each specialization.

4.9.3.2 Key_performance_indicator

The Key_performance_indicator class defines customer or user specific values per LSA_candidate. The respective value can be eg specified, contracted, allocated, distributed or actual.

Requirements are often stated as product design and performance data in the contract.

Key performance indicators can be defined for LSA Candidates at any level of indenture.

An instance of class Key_performance_indicator has no meaning by itself, but only in the context of a specific LSA_candidate.

Key_performance_indicator is an abstract class, ie an instantiation of Key_performance_indicator needs to be either:

- Product_service_life
- Scheduled_maintenance_interval
- Maintenance_free_operating_period
- Down_time
- Maintenance_man_hours_per_operating_hour
- Mean_time_between_unscheduled_removal
- Mean_time_to_repair
- Direct_maintenance_cost
- Shop_processing_time
- Failures_per_operating_hour
- Replacement_time
- Mean_time_between_failure
- Failure_rate

Key_performance_indicator attributes:

- Key_performance_indicator_type
- Key_performance_indicator_method (zero or one)
Key_performance_indicator_status
Key_performance_indicator_percintile (zero or one)

Note
The Key_performance_indicator class does not contain any key performance indicator values. These are contained within the respective subclass of Key_performance_indicator, eg Product_service_life.

Key_performance_indicator associations:
- An instance of Key_performance_indicator is always defined for a specific LSA_candidate
- An association with zero, one or many Operators (ie customers or users) that approved the Key_performance_indicator

Key_performance_indicator special recommendations:
- Where an instance of Key_performance_indicator is dependent upon eg operational environment, it must be distinguished by the assignment of an Applicability_statement to the Key_performance_indicator instance (also refer to Para 4.21, UoF Applicability Statement).

Note
An LSA Candidate can have more than one instance of the same Key_performance_indicator (eg Mean_time_between_failure). These values can be defined during different stages of the LSA candidate life cycle, eg one and the same LSA candidate can have one ‘specified’ mean time between failure value, and one ‘actual’ mean time between failure value.

4.9.3.3 Product_service_life
The Product_service_life class is a specialization of Key_performance_indicator, and represents the number of years the LSA Candidate is expected to be in service.

Product_service_life attributes:
- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Product_service_life (the value itself)

Product_service_life associations:
- An instance of Product_service_life is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator).
- An association with zero, one or many Operators (ie customers or users) that approved the Product_service_life value (inherited from class Key_performance_indicator).

Product_service_life special recommendations:
- An LSA candidate can have multiple defined Product_service_life's depending on operational environment, etc. If multiple Product_service_life’s are defined for the same type of key performance indicator (eg ‘contracted’), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed and contracted values for the Product_service_life attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to ‘minimum’. However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid.
- Representation of actual values for Product_service_life is recommended to use the Value_with_unit representation.
4.9.3.4 Scheduled_maintenance_interval
The Scheduled_maintenance_interval class is a specialization of Key_performance_indicator, and represents the (minimum) number of operational units (eg rounds, miles, hours) between scheduled maintenance.

Scheduled_maintenance_interval attributes:
- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Scheduled_maintenance_interval (the value itself)

Scheduled_maintenance_interval associations:
- An instance of Scheduled_maintenance_interval is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator)
- An association with zero, one or many Operators (ie customers or users) that approved the Scheduled_maintenance_interval value (inherited from class Key_performance_indicator)

Scheduled_maintenance_interval special recommendations:
- An LSA_candidate can have multiple defined Scheduled_maintenance_intervals depending on operational environment, etc. If multiple Scheduled_maintenance_intervals are defined for the same type of key performance indicator (eg "contracted"), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed and contracted values for the Scheduled_maintenance_interval attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to "minimum". However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Scheduled_maintenance_interval is recommended to use the Value_with_unit representation.

4.9.3.5 Maintenance_free_operating_period
The Maintenance_free_operating_period class is a specialization of Key_performance_indicator, and represents the acceptable (minimum) maintenance free operating period, where maintenance free operating period is the interval in which no maintenance actions occur.

Maintenance_free_operating_period attributes:
- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Maintenance_free_operating_period (the value itself)

Maintenance_free_operating_period associations:
- An instance of Maintenance_free_operating_period is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator).
- An association with zero, one or many Operators (ie customers or users) that approved the Maintenance_free_operating_period value (inherited from class Key_performance_indicator).

Maintenance_free_operating_period special recommendations:
- An LSA_candidate can have multiple defined Maintenance_free_operating_periods depending on operational environment, etc. If multiple Maintenance_free_operating_periods are defined for the same type of key performance indicator (eg 'contracted'), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed, and contracted values for the Maintenance_free_operating_period attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to ‘minimum’. However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Maintenance_free_operating_period is recommended to use the Value_with_unit representation.

4.9.3.6 Down_time
The Down_time class is a specialization of Key_performance_indicator, and represents the acceptable (maximum) down time (MDT), where MDT is the time where an item is non-operational.

Down_time attributes:
- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Down_time (the value itself)

Down_time associations:
- An instance of Down_time is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator).
- An association with zero, one or many Operators (ie customers or users) that approved the Down_time value (inherited from class Key_performance_indicator).

Down_time special recommendations:
- An LSA_candidate can have multiple defined Down_times depending on operational environment, etc. If multiple Down_times are defined for the same type of key performance indicator (eg ‘contracted’), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed and contracted values for the Down_time attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to ‘maximum’. However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Down_time is recommended to use the Value_with_unit representation.

4.9.3.7 Maintenance_man_hours_per_operating_hour
The Maintenance_man_hours_per_operating_hour class is a specialization of Key_performance_indicator, and represents the (maximum) acceptable maintenance man hours per operating hour, where maintenance man hours per operating hour is the ratio of maintenance man-hours expended to the operating interval (as defined by the measurement base) of the system/equipment.

Maintenance_man_hours_per_operating_hour attributes:
- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- **Maintenance_man_hours_per_operating_hour** (the value itself)

**Maintenance_man_hours_per_operating_hour associations:**

- An instance of Maintenance_man_hours_per_operating_hour is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator).
- An association with zero, one or many Operators (ie customers or users) that approved the Maintenance_man_hours_per_operating_hour value (inherited from class Key_performance_indicator).

**Maintenance_man_hours_per_operating_hour special recommendations:**

- An LSA_candidate can have multiple defined Maintenance_man_hours_per_operating_hours depending on operational environment, etc. If multiple Maintenance_man_hours_per_operating_hours are defined for the same type of key performance indicator (eg ‘contracted’), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed and contracted values for the Maintenance_man_hours_per_operating_hour attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to ‘maximum’. However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Maintenance_man_hours_per_operating_hour is recommended to use the Value_with_unit representation.

**4.9.3.8 Mean_time_between_unscheduled_removal**

The Mean_time_between_unscheduled_removal class is a specialization of Key_performance_indicator, and represents the total number of operational units (eg, miles, rounds, hours) divided by the total number of items removed from that system during a stated period of time. This term is defined to exclude removals as either being scheduled or performed in order to facilitate other maintenance and removals for product improvement.

**Mean_time_between_unscheduled_removal attributes:**

- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Mean_time_between_unscheduled_removal (the value itself)

**Mean_time_between_unscheduled_removal associations:**

- An instance of Mean_time_between_unscheduled_removal is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator).
- An association with zero, one or many Operators (ie customers or users) that approved the Mean_time_between_unscheduled_removal value (inherited from class Key_performance_indicator).

**Mean_time_between_unscheduled_removal special recommendations:**

- An LSA_candidate can have multiple defined Mean_time_between_unscheduled_removers depending on operational environment, etc. If multiple Mean_time_between_unscheduled_removers are defined for the same type of key performance indicator (eg ‘contracted’), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed and contracted values for the Mean_time_between_unscheduled_removal attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to ‘minimum’. However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for
4.9.3.9 Mean_time_to_repair

The Mean_time_to_repair class is a specialization of Key_performance_indicator, and represents the total elapsed time for corrective maintenance divided by the total number of corrective maintenance actions during a given period of time.

Mean_time_to_repair attributes:

- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Mean_time_to_repair (the value itself)

Mean_time_to_repair associations:

- An instance of Mean_time_to_repair is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator).
- An association with zero, one or many Operators (ie customers or users) that approved the Mean_time_to_repair value (inherited from class Key_performance_indicator).

Mean_time_to_repair special recommendations:

- An LSA_candidate can have multiple defined Mean_time_to_repair values depending on operational environment, etc. If multiple Mean_time_to_repair values are defined for the same type of key performance indicator (eg 'contracted'), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed and contracted values for the Mean_time_to_repair attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to 'maximum'. However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Mean_time_to_repair is recommended to use the Value_with_unit representation.

4.9.3.10 Direct_maintenance_cost

The Direct_maintenance_cost class is a specialization of Key_performance_indicator, and represents costs which include the shop maintenance man-hours, shop test man-hours, repair cost (incl. required material). Statistical values which represent the occurrence rate like MTBF (Mean Time Between Failure) and MTBUR (Mean Time Between Unscheduled Removal) must be considered for that cost element.

Direct_maintenance_cost attributes:

- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Direct_maintenance_cost (the value itself)

Direct_maintenance_cost associations:

- An instance of Direct_maintenance_cost is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator).
- An association with zero, one or many Operators (ie customers or users) that approved the Direct_maintenance_cost value (inherited from class Key_performance_indicator).

Mean_time_to_repair special recommendations:
− An LSA_candidate can have multiple defined Direct_maintenance_costs depending on operational environment, etc. If multiple Direct_maintenance_costs are defined for the same type of key performance indicator (eg ‘contracted’), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
− Representations of specified, allocated, distributed and contracted values for the Direct_maintenance_cost attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to 'maximum'. However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Direct_maintenance_cost is recommended to use the Value_with_unit representation.

4.9.3.11 Shop_processing_time
The Shop_processing_time class is a specialization of Key_performance_indicator, and represents the duration from the start of the repair activities in the repair shop until the closing of the repair procedure without considering any shipping and delay times.

Shop_processing_time attributes:
− Key_performance_indicator_type (inherited from class Key_performance_indicator)
− Key_performance_indicator_method (inherited from class Key_performance_indicator)
− Key_performance_indicator_status (inherited from class Key_performance_indicator)
− Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
− Shop_processing_time (the value itself)

Shop_processing_time associations:
− An instance of Shop_processing_time is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator)
− An association with zero, one or many operators (ie customers or users) that approved the Shop_processing_time value (inherited from class Key_performance_indicator).

Shop_processing_time special recommendations:
− An LSA_candidate can have multiple defined Shop_processing_times depending on operational environment, etc. If multiple Shop_processing_times are defined for the same type of key performance indicator (eg "contracted"), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
− Representations of specified, allocated, distributed and contracted values for the Shop_processing_time attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to "maximum". However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Shop_processing_time is recommended to use the Value_with_unit representation.

4.9.3.12 Failures_per_operating_hour
The Failures_per_operating_hour class is a specialization of Key_performance_indicator, and represents the failure rate, expressed in failures per operating hour.

The Failures_per_operating_hour defines a specified value which will be the maximum acceptable failure behavior of a component..

Failures_per_operating_hour attributes:
− Key_performance_indicator_type (inherited from class Key_performance_indicator)
− Key_performance_indicator_method (inherited from class Key_performance_indicator)
− Key_performance_indicator_status (inherited from class Key_performance_indicator)
− Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
− Failures_per_operating_hour (the value itself)
Failures_per_operating_hour associations:

- An instance of Failures_per_operating_hour is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator).
- An association with zero, one or many operators (ie customers or users) that approved the Failures_per_operating_hour value (inherited from class Key_performance_indicator).

Failures_per_operating_hour special recommendations:

- An LSA_candidate can have multiple defined Failures_per_operating_hour depending on operational environment, etc. If multiple Failures_per_operating_hour are defined for the same type of key performance indicator (eg "contracted"), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability statement.
- Representations of specified, allocated, distributed and contracted values for the Failures_per_operating_hour attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to "maximum". However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Failures_per_operating_hour is recommended to use the Value_with_unit representation.

4.9.3.13 Replacement_time

The Replacement_time class is a specialization of Key_performance_indicator, and represents the duration of the replacement of a (eg faulty) component within any technical system by another (eg new) component.

Note

The contracted/allocated/specified Replacement_time defines a specified value which will be the maximum acceptable duration for component replacement. In practice, this value can be used to document eg a customer's requirement to be able to perform 98% of all replacement tasks below a specified value of two hours (= maximum replacement time).

Replacement_time attributes:

- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Replacement_time (the value itself)

Replacement_time associations:

- An instance of Replacement_time is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator)
- An association with zero, one or many operators (ie customers or users) that approved the Replacement_time value (inherited from class Key_performance_indicator).

Replacement_time special recommendations:

- An LSA_candidate can have multiple defined Replacement_times depending on operational environment, etc. If multiple Replacement_times are defined for the same type of key performance indicator (eg "contracted"), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability statement.
- Representations of specified, allocated, distributed and contracted values for the Replacement_time attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to "maximum". However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Replacement_time is recommended to use the Value_with_unit representation.
4.9.3.14 Mean_time_between_failure

The Mean_time_between_failure (MTBF) class is a specialization of Key_performance_indicator, where the MTBF is the total operational life of a population of an LSA_Candidate divided by the total number of failures within the population during a particular measurement interval. The definition holds for time, rounds, miles, events, or other measure of life units.

Mean_time_between_failure attributes:
- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Mean_time_between_failure (the value itself)

Mean_time_between_failure associations:
- An instance of Mean_time_between_failure is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator)
- An association with zero, one or many operators (ie customers or users) that approved the Mean_time_between_failure value (inherited from class Key_performance_indicator)
- An optional association with zero, one or many instances of Mean_time_between_failure_correction.

Mean_time_between_failure special recommendations:
- An LSA_candidate can have multiple defined Mean_time_between_failures depending on operational environment, etc. If multiple Mean_time_between_failures are defined for the same type of key performance indicator (eg "contracted"), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed and contracted values for the Mean_time_between_failure attribute are recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to ‘minimum’. However, there can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Mean_time_between_failure is recommended to use the Value_with_unit representation.

4.9.3.15 Mean_time_between_failure_correction

The Mean_time_between_failure_correction class defines correction factors for an associated mean time between failure value. Mean_time_between_failure_correction can be defined from the following perspectives:
- Usage of the associated LSA_candidate under specific conditions, eg environment which causes additional stress to the system (sand, extreme temperatures, salty environment)
- Usage of a Part at a specific location in a Product Breakdown.

Mean_time_between_failure_correction attributes:
- Mean_time_between_failure_correction_factor
- Mean_time_between_failure_correction_justification
- Mean_time_between_failure_correction_date

The Mean_time_between_failure_correction class must implement the following interfaces:
- Applicability_assignment (Refer to UoF Applicability Statement).

Mean_time_between_failure_correction association:
- An instance of Mean_time_between_failure_correction is always associated with a specific instance of Mean_time_between_failure.
Mean_time_between_failure_correction special recommendations:

- The Mean_time_between_failure_correction is dependent upon eg operational environment, its location within a Product etc. This condition must be made explicit by the assignment of an Applicability_statement, as defined in UoF Applicability Statement.

4.9.3.16 Failure_rate

The Failure_rate class is a specialization of Key_performance_indicator, where the failure rate is the total number of failures within a population of an LSA_candidate item divided by the total functional life of the population during the measurement interval. The definition holds for time, rounds, miles, cycles or other measures of life units..

Failure_rate attributes:

- Key_performance_indicator_type (inherited from class Key_performance_indicator)
- Key_performance_indicator_method (inherited from class Key_performance_indicator)
- Key_performance_indicator_status (inherited from class Key_performance_indicator)
- Key_performance_indicator_percintile (inherited from class Key_performance_indicator)
- Failure_rate (the value itself)

Failure_rate associations:

- An instance of Failure_rate is always defined for a specific LSA_candidate (inherited from class Key_performance_indicator)
- An association with zero, one or many operators (ie customers or users) that approved the Failure_rate value (inherited from class Key_performance_indicator)
- An optional association with zero, one or many instances of Failure_rate_correction.

Failure_rate special recommendations:

- An LSA_candidate can have multiple defined Failure_rates depending on operational environment, etc. If multiple Failure_rates are defined for the same type of key performance indicator (eg "contracted"), these must be distinguished by the assignment of Applicability_statement as defined in UoF Applicability Statement.
- Representations of specified, allocated, distributed and contracted values for the Failure_rate attribute are recommended to use the Value_with_unit representation. However, there can also be cases where the use of Value_with_limit_property, Value_with_tolerances_property or Value_range_property is valid. Representation of actual values for Failure_rate is recommended to use the Value_with_unit representation.

4.9.3.17 Failure_rate_correction

The Failure_rate_correction class defines correction factors for an associated failure rate value. Failure_rate_correction can be defined from eg the following perspectives:

- Usage of the associated LSA_candidate under specific conditions, eg environment which causes additional stress to the system (sand, extreme temperatures, salty environment)
- Usage of a Part at a specific location in a Product Breakdown.

Failure_rate_correction attributes:

- Failure_rate_correction_factor

The Failure_rate_correction class must implement the following interfaces:

- Applicability_assignment (Refer to UoF Applicability Statement).

Failure_rate_correction associations:

- An instance of Failure_rate_correction is always associated with a specific instance of Failure_rate.
Failure_rate_correction special recommendations:

- The Failure_rate_correction is dependent upon eg operational environment, its location within a Product etc. This condition must be made explicit by the assignment of an Applicability_statement, as defined in UoF Applicability Statement.

4.9.4 UoF LSA Candidate - Additions to referenced class and interface definitions

4.9.4.1 Part
The Part class defined in UoF Part, must also implement the following additional interface:

- LSA_candidate

4.9.4.2 Breakdown_element_revision
The Breakdown_element_revision class defined in UoF Breakdown Structure, must also implement the following additional interface:

- LSA_candidate

4.9.4.3 Operator
The operator interface defined in UoF Project, must also implement the following additional association:

- An optional association with zero, one or many instances of Key_performance_indicators

4.10 UoF LSA Candidate Analysis Activity

4.10.1 Overall description
The LSA Candidate Analysis Activity UoF supports the selection and justification of Logistics Support Analysis (LSA) activities to be performed for the respective LSA Candidate.

Note
LSA Candidate analysis activities are only to be documented for those parts and Breakdown_element_revisions that are identified as being LSA Candidates.

Note
There is no requirement that all types of LSA activities must be performed for every single LSA Candidate, but there is a requirement to document the decision made for each combination of LSA Candidate and LSA activity.
4.10.2 Graphical representation

Fig 25 LSA Candidate Analysis Activity - class model
4.10.3 UoF LSA Candidate Analysis Activity - New class and interface definitions

4.10.3.1 Candidate_item_analysis_activity

The Candidate_item_analysis_activity class identifies the LSA activities that are to be performed for the respective LSA Candidate, the rationale behind it, and the recorded status for the respective analysis activity.

Each identified LSA activity can also be associated with zero, one or many documents, eg documents containing the results from the LSA activity.

**Note**
LSA activities are often selected during the LSA guidance conference.

**Note**
LSA activities can be defined for LSA Candidates at any level of indenture.

**Note**
The Candidate_item_analysis_activity class is an abstract class, ie the class itself will never be instantiated but any of its subclasses can be. Subclasses of Candidate_item_analysis_activity are:

- LSA_candidate_comparative_analysis_activity
- LSA_candidate_human_factor_analysis_activity
- LSA_candidate_reliability_analysis_activity
- LSA_candidate_maintainability_analysis_activity
- LSA_candidate_testability_analysis_activity
- LSA_candidate_failure_mode_and_effect_analysis_activity
- LSA_candidate_damage_analysis_activity
- LSA_candidate_special_event_analysis_activity
- LSA_candidate_scheduled_maintenance_analysis_activity
- LSA_candidate_level_of_repair_analysis_activity
- LSA_candidate_mainenance_task_analysis_activity
- LSA_candidate_software_data_loading_analysis_activity
- LSA_candidate_software_support_analysis_activity
- LSA_candidate_operational_analysis_activity
- LSA_candidate_simulation_operational_scenarios_analysis_activity
- LSA_candidate_training_needs_analysis_activity
- LSA_candidate_other_analysis_activity

Candidate_item_analysis_activity attributes:
- Candidate_item_analysis_activity_indicator
- Candidate_item_analysis_activity_rationale
- Candidate_item_analysis_activity_status
- Candidate_item_analysis_activity_date

Any subclass of Candidate_item_analysis_activity class must also implement the following interface:
- Document_assignment_interface

4.10.3.2 LSA_candidate_comparative_analysis_activity

The LSA_candidate_comparative_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding comparative analysis for the LSA Candidate under consideration.

**Note**
An instance of class LSA_candidate_comparative_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.
LSA_candidate_comparative_analysis_activity attributes:

- Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_comparative_analysis_activity associations:

- An association with the LSA_candidate for which the LSA_candidate_comparative_analysis_activity is being performed.

The LSA_candidate_comparative_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):

- Document_assignment_interface.

4.10.3.3 LSA_candidate_human_factor_analysis_activity

The LSA_candidate_human_factor_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding human factor analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_human_factor_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_human_factor_analysis_activity attributes:

- Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_human_factor_analysis_activity associations:

- An association with the LSA_candidate for which the LSA_candidate_human_factor_analysis_activity is being performed.

The LSA_candidate_human_factor_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):

- Document_assignment_interface.

4.10.3.4 LSA_candidate_reliability_analysis_activity

The LSA_candidate_reliability_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding reliability analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_reliability_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_reliability_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_reliability_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_reliability_analysis_activity is being performed

The LSA_candidate_reliability_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.5 LSA_candidate_maintainability_analysis_activity
The LSA_candidate_maintainability_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding maintainability analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_maintainability_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_maintainability_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_maintainability_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_maintainability_analysis_activity is being performed

The LSA_candidate_maintainability_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.6 LSA_candidate_testability_analysis_activity
The LSA_candidate_testability_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding testability analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_testability_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_testability_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_testability_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_testability_analysis_activity is being performed

The LSA_candidate_testability_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.7 LSA_candidate_failure_mode_and_effect_analysis_activity
The LSA_candidate_failure_mode_and_effect_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding failure mode and effect analysis (LSA-FMEA) for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_failure_mode_and_effect_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_failure_mode_and_effect_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_failure_mode_and_effect_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_failure_mode_and_effect_analysis_activity is being performed

The LSA_candidate_failure_mode_and_effect_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.8 LSA_candidate_damage_analysis_activity
The LSA_candidate_damage_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding damage analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_damage_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_damage_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_damage_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_damage_analysis_activity is being performed

The LSA_candidate_damage_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.9 LSA_candidate_special_event_analysis_activity
The LSA_candidate_special_event_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding special events analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_special_event_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_special_event_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_special_event_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_special_event_analysis_activity is being performed

The LSA_candidate_special_event_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.10 LSA_candidate_scheduled_maintenance_analysis_activity
The LSA_candidate_scheduled_maintenance_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding scheduled maintenance analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_scheduled_maintenance_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_scheduled_maintenance_analysis_activity attributes:
Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_scheduled_maintenance_analysis_activity associations:
- An association with the LSA_candidate for which the LSA_candidate_scheduled_maintenance_analysis_activity is being performed.

The LSA_candidate_scheduled_maintenance_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
- Document_assignment_interface.

4.10.3.11 LSA_candidate_level_of_repair_analysis_activity
The LSA_candidate_level_of_repair_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding level of repair analysis (LORA) for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_level_of_repair_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_level_of_repair_analysis_activity attributes:
- Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
- Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_level_of_repair_analysis_activity associations:
- An association with the LSA_candidate for which the LSA_candidate_level_of_repair_analysis_activity is being performed.

The LSA_candidate_level_of_repair_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
- Document_assignment_interface.

4.10.3.12 LSA_candidate_mainenance_task_analysis_activity
The LSA_candidate_mainenance_task_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding maintenance task analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_mainenance_task_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_mainenance_task_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_maintenance_task_analysis_activity associations:
− An association with the LSA_candidate for which the
  LSA_candidate_maintenance_task_analysis_activity is being performed

The LSA_candidate_maintenance_task_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.13 LSA_candidate_software_data_loading_analysis_activity

The LSA_candidate_software_data_loading_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding software data loading analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_software_data_loading_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_software_data_loading_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_software_data_loading_analysis_activity associations:
− An association with the LSA_candidate for which the
  LSA_candidate_software_data_loading_analysis_activity is being performed

The LSA_candidate_software_data_loading_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.14 LSA_candidate_software_support_analysis_activity

The LSA_candidate_software_support_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding software support analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_software_support_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_software_support_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_software_support_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_software_support_analysis_activity is being performed

The LSA_candidate_software_support_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.15 LSA_candidate_operational_analysis_activity
The LSA_candidate_operational_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding operational analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_operational_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_operational_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_operational_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_operational_analysis_activity is being performed

The LSA_candidate_operational_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.16 LSA_candidate_simulation_operational_scenarios_analysis_activity
The LSA_candidate_simulation_operational_scenarios_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding simulation of operational scenarios for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_simulation_operational_scenarios_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_simulation_operational_scenarios_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_simulation_operational_scenarios_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_simulation_operational_scenarios_analysis_activity is being performed

The LSA_candidate_simulation_operational_scenarios_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.17 LSA_candidate_training_needs_analysis_activity
The LSA_candidate_training_needs_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding training needs analysis for the LSA Candidate under consideration.

Note
An instance of class LSA_candidate_training_needs_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_training_needs_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_training_needs_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_training_needs_analysis_activity is being performed

The LSA_candidate_training_needs_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.3.18 LSA_candidate_other_analysis_activity
The LSA_candidate_other_analysis_activity class is a specialization of Candidate_item_analysis_activity, and records the decision made regarding additional types of analysis activities that needs to be performed for the LSA Candidate under consideration, apart from those enumerated above.

Note
An instance of class LSA_candidate_other_analysis_activity has no meaning by itself, but only in the context of a specific LSA_candidate.

LSA_candidate_other_analysis_activity attributes:
− Candidate_item_analysis_activity_indicator (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_rationale (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_status (inherited from class Candidate_item_analysis_activity)
− Candidate_item_analysis_activity_date (inherited from class Candidate_item_analysis_activity)

LSA_candidate_other_analysis_activity associations:
− An association with the LSA_candidate for which the LSA_candidate_other_analysis_activity is being performed

The LSA_candidate_other_analysis_activity class must also implement the following interface (inherited from class Candidate_item_analysis_activity):
− Document_assignment_interface.

4.10.4 UoF LSA Candidate Analysis Activity - Additions to referenced class and interface definitions

4.10.4.1 LSA Candidate
Any class that implements the LSA_candidate interface defined in UoF LSA Candidate must also implement the following additional associations:
− An association with LSA_candidate_comparative_analysis_activity
− An association with LSA_candidate_human_factor_analysis_activity
− An association with LSA_candidate_reliability_analysis_activity
− An association with LSA_candidate_maintainability_analysis_activity
− An association with LSA_candidate_reliability_analysis_activity
− An association with LSA_candidate_testability_analysis_activity
− An association with LSA_candidate_failure_mode_and_effect_analysis_activity
− An association with LSA_candidate_damage_analysis_activity
− An association with LSA_candidate_special_event_analysis_activity
− An association with LSA_candidate_scheduled_maintenance_analysis_activity
− An association with LSA_candidate_level_of_repair_analysis_activity
− An association with LSA_candidate_mainenance_task_analysis_activity
− An association with LSA_candidate_software_data_loading_analysis_activity
− An association with LSA_candidate_software_support_analysis_activity
− An association with LSA_candidate_operational_analysis_activity
− An association with LSA_candidate_simulation_operational_scenarios_analysis_activity
− An association with LSA_candidate_training_needs_analysis_activity
− An association with LSA_candidate_other_analysis_activity

Note
Only those instances of Breakdown_element_revision and Part (ie those classes that implement the LSA_candidate interface) that are selected as LSA Candidates must identify which analysis activities are to be performed.

4.11 UoF LSA-FMEA and Special Events

4.11.1 Overall description
The LSA-FMEA and Special Events UoF supports the recording of portions of the results from the following LSA analysis activities:
− LSA-FMEA (Failure Mode and Effect Analysis),
− Damage analysis
− Special event analysis
− Testability analysis
4.11.1.1 LSA-FMEA
The starting point for the LSA-FMEA are the physical items that has been identified as being LSA candidates. Physical items can be Hardware_element_revisions, as well as Hardware_parts, and each specific physical item can have a set of identified failure modes.

All identified failure modes causing one and the same maintenance procedure will be grouped into a single LSA failure mode. Each LSA failure mode will be associated with the hardware element or part for which there will be a defined task requirement, where the task requirement represents the need for the identified maintenance procedure. This hardware element or part need not be the same hardware element or part as for which the original failure mode(s) was identified.

While grouping, failure rates for the item under analysis is distributed to the respective LSA failure mode. The distributed failure rate can be defined either by ratio, or by qualitative measures (rating).

Each possible failure mode can be detected by, either an automatic detection mean, BIT (Built-In-Test), or by other functional/physical symptoms. The failure mode effects defined in the technical FME(C)A can be considered as a guideline. The ability to localize the unit that has failed (without ambiguity) can be rated using qualitative measures.

The requirement for a troubleshooting task can be recorded against the original failure mode.

4.11.1.2 Damage and special events
Special events can be due to external or internal causes, eg natural phenomenon or unusual use. The probability of an event occurring is related to the usage phase of the product, eg operation, maintenance etc.

Each possible event can affect one or many LSA candidates and result in one or many failure modes. Identification of potential damages can also include a special analysis for each affected LSA candidate with respect to the knowledge of the technology used. This is referred to as technology behavior rating.
4.11.2 Graphical representation

Fig 26 LSA-FMEA and Special Events - class model

ICN-B6865-S3000L0225-002-01

Applicable to: All
4.11.3 UoF LSA-FMEA and Special Events - New class and interface definitions

4.11.3.1 Physical_item
The Physical_item interface is implemented by classes that represent hardware items.

Classes that implement the Physical_item interface:

- Hardware_element_revision
- Hardware_part

Classes that implement the Physical_item interface must implement the following associations:

- An optional association with one or many instances of Failure_mode
- An optional association with one or many instances of LSA_failure_mode

4.11.3.2 Failure_mode
The Failure_mode class supports the definition of possible failure modes for a specific Physical_item.

**Note**
An instance of class Failure_mode has no meaning by itself, but only in the context of a Physical_item.

**Note**
Failure mode is also known as damage for failure modes caused by special events.

Failure_mode attributes:

- Failure_mode_identifier (zero or one)
- Failure_mode_description
- Failure_mode_detection_ability_rating (zero or one)
- Failure_mode_detection_ability_description (zero or one)
- Failure_mode_localization_ability_rating (zero or one)
- Failure_mode_localization_ability_description (zero or one)

**Note**
Failure_mode_detection_ability_rating and Failure_mode_localization_ability_rating are both of data type Classification, i.e., both ratings must have an associated date representing the date when the rating was performed.

LSA_candidate_failure_mode associations:

- An association with the Physical_item for which the failure mode has been identified
- An association with one or many instances of Failure_mode_effect
- An optional association with one or many Special_events that can cause the identified failure mode (via the Special_event_effects class)
- An optional association with one or many Detection_means that can be used to detect the failure mode
- An optional association with one LSA_failure_mode (grouping of failure modes)
- An optional association with a troubleshooting Task_requirement

4.11.3.3 LSA_failure_mode
The LSA_failure_mode class supports the grouping of identified failure modes which leads to the same maintenance procedure.

**Note**
The LSA_failure_mode class is an abstract class, i.e., an instantiation of LSA_failure_mode needs to be either:
• Distributed_LSA_failure_mode_with_ratio
• Distributed_LSA_failure_mode_with_rating

Making the LSA_failure_mode an abstract class, mandates the distribution of failure rates
between the different instances of LSA_failure_mode that has been identified for the same
Physical_item.

LSA_failure_mode attributes:

− LSA_failure_mode_identifier
− LSA_failure_mode_description (zero or one)

LSA_failure_mode associations:

− An association with the Physical_item for which the LSA_failure_mode has been identified
− An association with one or many instances of Failure_mode which are grouped into the
  LSA_failure_mode
− An association with the Task_requirement that describes the identified maintenance
  procedure

4.11.3.4 Distributed_LSA_failure_mode_with_ratio

The Distributed_LSA_failure_mode_with_ratio class is a specialization of LSA_failure_mode,
and records the probability for an individual LSA_failure_mode to occur in relation to the entire
population of LSA_failure_modes identified for the related Physical_item.

Note
An instance of class Distributed_LSA_failure_mode_with_ratio has no meaning by itself,
but only in the context of a specific Physical_item.

Distributed_LSA_failure_mode_with_ratio attributes:

− LSA_failure_mode_identifier (inherited from class LSA_failure_mode)
− LSA_failure_mode_description (inherited from class LSA_failure_mode)
− LSA_failure_mode_distribution_ratio (one or many)

Distributed_LSA_failure_mode_with_ratio associations:

− An association with the Physical_item for which the
  Distributed_LSA_failure_mode_with_ratio has been identified (inherited from class
  LSA_failure_mode)
− An association with one or many instances of Failure_mode which are grouped into the
  Distributed_LSA_failure_mode_with_ratio (inherited from class LSA_failure_mode)
− An association with the Task_requirement that describes the identified maintenance
  procedure (inherited from class LSA_failure_mode)

Distributed_LSA_failure_mode_with_ratio special recommendations:

− A Distributed_LSA_failure_mode_with_ratio can have multiple failure mode distribution
  values, eg depending on operational environment. Each instance of
  LSA_failure_mode_distribution_ratio is then recommended have an associated
  Applicability_statement, as defined in the UoF Applicability Statement, in order to define
  when the respective ratio value is applicable.

4.11.3.5 Distributed_LSA_failure_mode_with_rating

The Distributed_LSA_failure_mode_with_rating class is a specialization of LSA_failure_mode,
and records the rating for the occurrence of an individual LSA_failure_mode in comparison to the
entire population of LSA failure modes identified for the related Physical_item.
Note
An instance of class Distributed_LSA_failure_mode_with_rating has no meaning by itself, but only in the context of a specific Physical_item.

Distributed_LSA_failure_mode_with_rating attributes:
- LSA_failure_mode_identifier (inherited from class LSA_failure_mode)
- LSA_failure_mode_description (inherited from class LSA_failure_mode)
- LSA_failure_mode_distribution_occurrence_rating (one or many)

Distributed_LSA_failure_mode_with_rating associations:
- An association with the Physical_item for which the Distributed_LSA_failure_mode_with_rating has been identified (inherited from class LSA_failure_mode)
- An association with one or many Instances of Failure_mode which are grouped into the Distributed_LSA_failure_mode_with_rating (inherited from class LSA_failure_mode)
- An association with the Task_requirement that describes the identified maintenance procedure (inherited from class LSA_failure_mode)

Distributed_LSA_failure_mode_with_rating special recommendations:
- A Distributed_LSA_failure_mode_with_rating can have multiple failure mode distribution ratings, eg depending on operational environment. Each instance of Distributed_LSA_failure_mode_with_rating is then recommended have an associated Applicability_statement, as defined in the UoF Applicability Statement, in order to define when the respective rating is applicable.

4.11.3.6 Failure_mode_effect
The Failure_mode_effect class defines the consequences of an identified failure mode and its effect on the local/next higher/end item operation, function, or status.

Note
An instance of class Failure_mode_effect has no meaning by itself, but only in the context of an identified Failure_mode.

Note
The Failure_mode_effect class is an abstract class, ie an instantiation of Failure_mode_effect must to be either:
- Local_failure_mode_effect
- Higher_failure_mode_effect

Failure_mode_effect attributes:
- Failure_mode_effect

Failure_mode_effect associations:
- An association with the Failure_mode that causes the identified Failure_mode_effect

4.11.3.7 Local_failure_mode_effect
The Local_failure_mode_effect class is a specialization of Failure_mode_effect, and identifies the consequences of each postulated failure/damage mode affecting the Physical_item under analysis. It is possible for the "local effect" to be the failure/damage mode itself.

Note
An instance of class Local_failure_mode_effect has no meaning by itself, but only in the context of an identified Failure_mode.

Local_failure_mode_effect attributes:
4.11.3.8 Higher_failure_mode_effect
The Higher_failure_mode_effect class is a specialization of Failure_mode_effect, and identifies the consequences of each failure/damage mode affecting the next higher indenture level, or the essential functions(s) affecting system/equipment operating capability and mission completion capability.

**Note**
An instance of class Higher_failure_mode_effect has no meaning by itself, but only in the context of an identified Failure_mode.

**Higher_failure_mode_effect attributes:**
- Failure_mode_effect (inherited from class Failure_mode_effect)
- Failure_mode_effect_level

**Higher_failure_mode_effect associations:**
- An association with one or many Instances of Failure_mode that causes the identified Higher_failure_mode_effect (inherited from class Failure_mode_effect)
- An association with the LSA_candidate that represents the higher indenture level

4.11.3.9 Detection_mean
The Detection_mean interface is implemented by those classes whose instances can be used for detection and/or localization of one or many failure modes.

**Classes that implement the Detection_mean interface:**
- Hardware_element_revision
- Hardware_part

**Detection_mean attributes:**
- Detection_mean_description
- Detection_mean_false_alarm_rate
- Detection_mean_type

**Detection_mean associations:**
- An association with one or many Failure_modes that can be detected/localized using the Detection_mean

4.11.3.10 Detectability
The Detectability class is an association class, which adds information to a defined relationship between a Detection_mean and an identified Failure_mode which can be detected and/or localized, using the detection mean.

**Detectability attributes:**
- Failure_mode_detection_rate (zero or one)
- Failure_mode_isolation_rate (zero or one)

4.11.3.11 Special_event
The Special_event class identifies special events that can occur, and which can lead to one or many failure modes for one or many affected LSA_candidates.
Special_event attributes:
- Special_event_title
- Special_event_description (zero or one)
- Special_event_group

Special_event associations:
- An optional association with one or many instances of LSA_candidate which can be affected by the special event
- An association with one or many instances of Special_event_occurrence_definition, which defines the probability for the special event to occur during a specific product usage phase.

4.11.3.12 Special_event_occurrence_definition
The Special_event_occurrence_definition class defines the probability for a special event to occur during a specific product usage phase.

Note
The Special_event_occurrence_definition class is an abstract class, i.e. an instantiation of Special_event_occurrence_definition must be either:
- Rated_special_event_occurrence_definition
- Quantified_special_event_occurrence_definition

Note
An instance of class Special_event_occurrence_definition has no meaning by itself, but only in the context of an identified Special_event.

Special_event_occurrence_definition associations:
- An association with one or many defined Product_usage_phases during which the Special_event can occur
- An association with the Special_event that can occur during the specified product usage phase

Special_event_occurrence_definition special recommendations:
- A Special_event_occurrence_definition can be dependent upon e.g. operational environment. Each instance of Special_event_occurrence_definition is then recommended to have an associated Applicability_statement, as defined in the UoF Applicability Statement, in order to define when the respective Special_event_occurrence_definition is applicable.

4.11.3.13 Product_usage_phase
The Product_usage_phase class defines usage phases of interest for the Product in scope of the LSA project.

Product_usage_phase attributes:
- Product_usage_phase

Product_usage_phase associations:
- An optional association with one or many Special_events that can occur during the specified Product_usage_phase (via Special_event_occurrence_definition)

4.11.3.14 Rated_special_event_occurrence_definition
The Rated_special_event_occurrence_definition class is a specialization of the Special_event_occurrence_definition class.
Rated_special_event_occurrence_definition and Quantified_special_event_occurrence_definition respectively, mandate the recording of the frequency for which the associated Special_event is expected to occur during a specific product usage phase, either by rating or by ratio.

Note: An instance of class Rated_special_event_occurrence_definition has no meaning by itself, but only in the context of an identified Special_event.

Rated_special_event_occurrence_definition attributes:
- Special_event_occurrence_rating

Rated_special_event_occurrence_definition associations:
- An association with one or many defined Product_usage_phases during which the Special_event can occur (inherited from class Special_event_occurrence_definition)
- An association with the Special_event that can occur during the specified product usage phase (inherited from class Special_event_occurrence_definition)

Rated_special_event_occurrence_definition special recommendations:
- A Rated_special_event_occurrence_definition can be dependent upon eg operational environment. Each instance of Rated_special_event_occurrence_definition is then recommended have an associated Applicability_statement, as defined in the UoF Applicability Statement, in order to define when the respective Rated_special_event_occurrence_definition is applicable.

4.11.3.15 Quantified_special_event_occurrence_definition
The Quantified_special_event_occurrence_definition class is a specialization of the Special_event_occurrence_definition class.

Note: An instance of class Quantified_special_event_occurrence_definition has no meaning by itself, but only in the context of an identified Special_event.

Quantified_special_event_occurrence_definition attributes:
- Special_event_occurrence_rate

Quantified_special_event_occurrence_definition associations:
- An association with one or many defined Product_usage_phases during which the Special_event can occur (inherited from class Special_event_occurrence_definition)
- An association with the Special_event that can occur during the specified product usage phase (inherited from class Special_event_occurrence_definition)

Quantified_special_event_occurrence_definition special recommendations:
- A Quantified_special_event_occurrence_definition can be dependent upon eg operational environment. Each instance of Quantified_special_event_occurrence_definition is then recommended have an associated Applicability_statement, as defined in the UoF Applicability Statement, in order to define when the respective Quantified_special_event_occurrence_definition is applicable.
4.11.3.16 Special_event_effects
The Special_event_effects class is an association class, which adds information to a defined relationship between a Special_event and an affected LSA_candidate.

Special_event_effects associations:
- An optional association with one or many identified Failure_modes that can occur as a consequence of the related Special_event.

Note
The probability for each associated Failure_mode to occur, is defined using the association class Special_event_effect_probability.

4.11.3.17 Special_event_effect_probability
The Special_event_effect_probability class is an association class, which adds information to a defined relationship between a Special_events_effects and an associated Failure_mode.

Special_event_effect_probability attributes:
- Special_event_effect_probability_ratio (zero or one)

4.11.3.18 LSA_candidate_technology_behavior_rating
The LSA_candidate_technology_behavior_rating class identifies technologies used and describes its characteristics regarding knowledge of its behavior. Experience shows that technologies used can be more or less sensitive to damage.

Note
An LSA Candidate can have zero, one or many technology behavior ratings over time.

Note
An instance of class LSA_candidate_technology_behavior_rating has no meaning by itself, but only in the context of an LSA_candidate.

LSA_candidate_technology_behavior_rating attributes:
- Technology_behavior_knowledge_rating (one or many)
- Technology_sensitivity_rating (one or many)

LSA_candidate_technology_behavior_rating association:
- An association with the LSA_candidate for which the technology behavior rating is being performed

4.11.4 UoF LSA-FMEA and Special Events - Additions to referenced class and interface definitions
4.11.4.1 Hardware_part
The Hardware_part class defined in UoF Part, must also implement the following additional interfaces:
- Physical_item
- Detection_mean

4.11.4.2 Hardware_element_revision
The Hardware_element_revision class defined in UoF Breakdown Element Realization must also implement the following additional interfaces:
- Physical_item
- Detection_mean

4.11.4.3 LSA_candidate
The LSA_candidate interface defined in UoF LSA Candidate, must also implement the following additional associations:
− An optional association with one or many instances of Higher_failure_mode_effect
− An optional association with one or many instances of Special_event that can affect the LSA_candidate
− An optional association with an instance of LSA_candidate_technology_behavior_rating

4.11.4.4 Task_requirement
The Task_requirement class defined in UoF LSA Candidate Task Requirement, must also implement the following additional associations:

− An optional association with an instance of LSA_failure_mode, defining a need for a rectifying maintenance procedure
− An optional association with one or many instances of Failure_mode, defining a need for a troubleshooting procedure

4.12 UoF LSA Candidate Task Requirement
4.12.1 Overall description
The LSA Candidate Task Requirement UoF supports early recordings of task requirements, which will be refined during the detailed maintenance task analysis, software and data loading analysis etc.

Objectives for documenting task requirements are, eg to:

− Produce input to the LSA review, and together with the customer define maintenance concepts for the respective LSA Candidate before any detailed task analysis is being carried out.
− Identify product design changes that will result in increased maintainability, testability, etc.

A task requirement can also be associated with the source for the requirement, ie the driver for the task. The requirement for a task may be derived from any type of LSA activity (Para 4.10).
4.12.2 Graphical representation

4.12.3 UoF LSA Candidate Task Requirement - New class and interface definitions
4.12.3.1 Task_requirement

The Task_requirement class supports the specification of task requirements prior to any detailed task analysis.

Task_requirement attributes:
- Task_requirement_identifier
- Task_requirement_revision_identifier
- Task_requirement_date
- Task_requirement_description
- Task_requirement_decision
- Task_requirement_special_resource (zero or one)

Task_requirement associations:
- An association with one or many LSA_candidates for which the task is required (via Task_requirement_target)
- An optional association with one or many succeeding revisions of the Task_requirement
- An optional association with one or many preceding revisions of the Task_requirement
- An optional association with one or many Candidate_item_analysis_activities, from which the Task_requirement is derived
4.12.3.2 Task_requirement_change
The Task_requirement_change class is an association class, which adds information to the association between two successive revisions of Task_requirement.

Task_requirement_change attribute:
- Task_requirement_change_description (zero or one)

4.12.3.3 Authority_driven_task_requirement
The Authority_driven_task_requirement class is a specialization of class Task_requirement, and is to be used to record task requirements which are derived from regulations and/or other authoritative sources.

Authority_driven_task_requirement attributes:
- Task_requirement_identifier (inherited from class Task_requirement)
- Task_requirement_revision_identifier (inherited from class Task_requirement)
- Task_requirement_date (inherited from class Task_requirement)
- Task_requirement_description (inherited from class Task_requirement)
- Task_requirement_decision (inherited from class Task_requirement)
- Task_requirement_special_resource (inherited from class Task_requirement)
- Task_requirement_authority_source_type (zero or one)

The Authority_driven_task_requirement class must also implement the following interface:
- Document_assignment (Refer to UoF Document)

Authority_driven_task_requirement associations:
- An association with one or many LSA_candidates for which the task is required (inherited from class Task_requirement)
- An optional association with one or many Candidate_item_analysis_activities, from which the task requirement is derived (inherited from class Task_requirement)
- An optional association with one or many product design Change_requests, which are required in order to be able to carry out the required task in the specified way (inherited from class Task_requirement)
- An association with the authority Organization.

Authority_driven_task_requirement special recommendations:
- Each instance of Authority_driven_task_requirement should have at least one document reference from which the task requirement is derived, using the Document_assignment capability defined in UoF document (refer to Para 4.19).

4.12.3.4 Task_requirement_target
The Task_requirement_target class defines a relationship between a specific Task_requirement and the LSA_candidate for which the task is required.

Note
There is one instance of Task_requirement_target per relevant combination of Task_requirement and LSA_candidate.

Task_requirement_target associations:
- An association with the Task_requirement which is required for the LSA Candidate
- An association with the LSA_candidate for which the task is required
An optional association with one or many Task_limits defining, eg a task limit as required by an authority

**Note**
The Task_limit class is defined in UoF Task Usage Part 1. Refer to Para 4.15.

**4.12.3.5** Change_request
The Change_request class supports the recording of desired product design changes.

Change_request attributes:
- Change_request_identifier (zero or one)
- Change_request_description

Change_request associations:
- An association with one or many instances of Task_requirement from which the design change request has been derived
- An association with the LSA_candidate for which the design change request is defined

**4.12.4** UoF LSA Candidate Task Requirement - Additions to referenced class and interface definitions

**4.12.4.1** LSA_candidate
The LSA_candidate interface defined in UoF LSA Candidate, must also implement the following additional associations:
- An optional association with one or many Task_requirements (via Task_requirement_target)
- An optional association with one or many Change_requests

**4.12.4.2** Candidate_item_analysis_activity
The Candidate_item_analysis_activity class defined in UoF LSA Candidate Analysis Activity, must also implement the following additional association:
- An optional association with one or many Task_requirements, which refer to the Candidate_item_analysis_activity as its justification

**4.12.4.3** Organization
The Organization class defined in UoF Project, must also implement the following additional association:
- An optional association with one or many instances of Authority_driven_task_requirement, in the role of authority organization.

**4.12.4.4** Task_limit
The Task_limit class defined in UoF Task Usage (Part 1), must also implement the following additional association:
- An association with an instance of Task_requirement_target, for which the Task_limit can be defined.

**4.13** UoF Task

**4.13.1** Overall description
The Task UoF supports a detailed definition of a task, both maintenance task and operational task.

The Task procedure is described using a set of Subtasks. A Subtask can either be defined and described within the Task under consideration, or make reference to another Task.

**Note**
A Subtask that is defined and described within a Task cannot be referenced by any other Task.
There is also the possibility to define a time line (schedule) for the Subtasks contained within a Task. This can be used for the purpose of optimizing resource requirements and/or be the basis for creating job cards, eg in a fleet management system.

**Note**

The Task UoF has been designed to support an easy to use integration between S3000L and the S1000D procedure and schedule schemes, respectively.
4.13.2 Graphical representation

Fig 28 UoF Task - class model
4.13.3 UoF Task - New class and interface definitions

4.13.3.1 Task

The Task class supports the detailed specification of a task, both maintenance tasks and operational tasks.

Note

Task is an abstract class, ie an instantiation of Task must be either, a Rectifying_task, Operational_task or a Supporting_task. (Refer to Chap 12 for definitions and usages of the respective task type).

Note

A detailed procedure for performing the task can be documented in zero, one or many S1000D data modules. The Task UoF supports cross references between tasks being defined in S3000L and data modules being produced in accordance with the S1000D procedure and schedule schemes, respectively.

Note

Task includes its revision identification.

Task attributes:

- Task_identifier (one or many)
- Task_revision_identifier
- Task_name (one or many)
- Task_revision_status (zero or one)
- Information_code
- Task_criticality_code (zero or one)
- Task_operability_code
- Task_duration (zero, one or many)
- Task_total_labour_time (zero, one or many)
- S1000D_task_type (zero or one)

Task associations:

- An optional association with zero, one or many succeeding revisions of the Task
- An optional association with zero, one or many preceding revisions of the Task
- An optional association with zero or one Organization that is responsible for the task definition (can be used if the task definition is imported from e.g. a subcontractor)
- An optional association with zero, one or many Warning_caution_or_notes (via Task_warning_caution_or_note)
- An optional association with zero, one or many S1000D data modules, which contains the detailed procedures for performing the task
- An optional association with zero, one or many Task_requirements which forms the justification for the task
- An association with one or many Subtasks which contains the details for the task
- An optional association with zero, one or many instances of Subtask_by_reference, i.e. Subtasks that reference the Task under consideration as a Subtask

Note

The association between Task and S1000D_data_module classes enables the definition of cross references between the tasks that are defined in S3000L, and the corresponding S1000D data modules.

A Task in S3000L can be split into many S1000D data modules. One S1000D data module can cover many S3000L tasks. However, projects are strongly recommended to aim for a one to one correspondence.
Where the concept of master data modules is used in the S1000D CSDB (Common Source Database), it is recommended to just keep the cross references between the S3000L tasks and the master data modules in the CSDB.

Task special recommendations:

- The value for the Task_duration and Task_total_labour_time attributes can be recommended to use either of the following representations:
  - Value_with_unit_property
  - Value_with_limit_property, where Limit_qualifier is set to "maximum".
  - Value_with_tolerances_property
  - Value_range_property

- Where a Task has different Task_durations and/or Task_total_labour_times depending on, eg maintenance level, these must be represented as a set of values (Prp instances) for the respective attribute, but be distinguished by assigning Applicability_statement to the respective Prp instance representing the individual value (also refer to Para 4.21, UoF Applicability Statement).

4.13.3.2 Task_change
The Task_change class is an association class which adds information to the association between two successive revisions of a Task.

Task_change attributes:

- Task_revision_change_description (zero or one)

4.13.3.3 Task_distribution
The Task_distribution class is an association class which adds information to the association between a Task_requirement and a Task. Task_distribution records the probability of one Task to be performed, when there are multiple Tasks that can resolve the same Task_requirement.

Task_distribution attributes:

- Task_distribution_ratio

4.13.3.4 Rectifying_task
The Rectifying_task class is a specialization of class Task.

Note
Each maintenance activity is driven by an event. This event can be a failure, damage, special event or a time limit (interval). All these events require a maintenance action that resolves the event. Each task that is able to resolve an event must be defined as a rectifying task.

Rectifying_task attributes:

- Task_identifier (inherited from class Task)
- Task_revision_identifier (inherited from class Task)
- Task_name (inherited from class Task)
- Task_revision_status (inherited from class Task)
- Information_code (inherited from class Task)
- Task_criticality_code (inherited from class Task)
- Task_operability_code (inherited from class Task)
- Task_duration (inherited from class Task)
- Task_total_labour_time (inherited from class Task)
- S1000D_task_type (inherited from class Task)
Rectifying_task associations:

- An optional association with zero, one or many succeeding revisions of the Rectifying_task (inherited from class Task)
- An optional association with zero, one or many preceding revisions of the Rectifying_task (inherited from class Task)
- An optional association with zero or one Organization that is responsible for the task definition (inherited from class Task)
- An optional association with zero or one many Warning_caution_or_notes (via Task_warning_caution_or_note) (inherited from class Task)
- An optional association with zero, one or many S1000D data modules, which contains the detailed procedures for performing the Rectifying_task (inherited from class Task)
- An optional association with zero, one or many Task_requirements which forms the justification for the Rectifying_task (inherited from class Task)
- An association with one or many Subtasks which contains the details for the Rectifying_task (inherited from class Task)
- An optional association with zero, one or many instances of Subtask_by_reference, ie Subtasks that uses the Rectifying_task under consideration as a subtask (inherited from class Task).

Rectifying_task special recommendations:

- See special recommendations described under Task

4.13.3.5 Supporting_task

The Supporting_task class is a specialization of class Task.

Note

A supporting task does not "solve" any event, but can be used as a subtask within one or many rectifying tasks. An example of a supporting task is open hatch, or jack a car.

Supporting_task attributes:

- Task_identifier (inherited from class Task)
- Task_revision_identifier (inherited from class Task)
- Task_name (inherited from class Task)
- Task_revision_status (inherited from class Task)
- Information_code (inherited from class Task)
- Task_criticality_code (inherited from class Task)
- Task_operability_code (inherited from class Task)
- Task_duration (inherited from class Task)
- Task_total_labour_time (inherited from class Task)
- S1000D_task_type (inherited from class Task)

Supporting_task associations:

- An optional association with zero, one or many succeeding revisions of the Supporting_task (inherited from class Task)
- An optional association with zero, one or many preceding revisions of the Supporting_task (inherited from class Task)
- An optional association with zero or one Organization that is responsible for the task definition (inherited from class Task)
- An optional association with zero or one many Warning_caution_or_notes (via Task_warning_caution_or_note) (inherited from class Task)
- An optional association with zero, one or many S1000D data modules, which contains the detailed procedures for performing the Supporting_task (inherited from class Task)
- An optional association with zero, one or many Task_requirements which forms the justification for the Supporting_task (inherited from class Task)
− An association with one or many Subtasks which contains the details for the Supporting_task (inherited from class Task)
− An optional association with zero, one or many instances of Subtask_by_reference, ie Subtasks that uses the Supporting_task under consideration as a subtask (inherited from class Task)

Supporting_task special recommendations:
− See special recommendations described under Task

4.13.3.6 Operational_task
The Operational_task class is a specialization of class Task.

Note
Operational tasks are tasks that are required for operational purposes, eg fueling. An operational task may also be used as a subtask within one or many rectifying tasks.

Operational_task attributes:
− Task_identifier (inherited from class Task)
− Task_revision_identifier (inherited from class Task)
− Task_name (inherited from class Task)
− Task_revision_status (inherited from class Task)
− Information_code (inherited from class Task)
− Task_criticality_code (inherited from class Task)
− Task_operability_code (inherited from class Task)
− Task_duration (inherited from class Task)
− Task_total_labour_time (inherited from class Task)
− S1000D_task_type (inherited from class Task)

Operational_task associations:
− An optional association with zero, one or many succeeding revisions of the Operational_task (inherited from class Task)
− An optional association with zero, one or many preceding revisions of the Operational_task (inherited from class Task)
− An optional association with zero or one Organization that is responsible for the task definition (inherited from class Task)
− An optional association with zero, one or many Warning_caution_or_notes (via Task_warning_caution_or_note) (inherited from class Task)
− An optional association with zero, one or many S1000D data modules, which contains the detailed procedures for performing the Operational_task (inherited from class Task)
− An optional association with zero, one or many Task_requirements which forms the justification for the Operational_task (inherited from class Task)
− An aggregate association with one or many Subtasks which contains the details for the Operational_task (inherited from class Task)
− An optional association with zero, one or many instances of Subtask_by_reference, ie Subtasks that uses the Operational_task under consideration as a subtask (inherited from class Task)

Operational_task special recommendations:
− See special recommendations described under Task

4.13.3.7 Subtask
The Subtask class defines steps to be performed within a Task. Subtasks are used to provide detailed information about the Task.
Note
Subtask is an abstract class, ie an instantiation of Subtask needs to be either, a Subtask_by_definition, a Subtask_by_reference, or a Subtask_by_external_reference.

Note
Subtasks within a Task can be time lined (scheduled) using the Subtask_timeline class.

Subtask attributes:
- Subtask_identifier
- Subtask_role (zero or one)

Subtask associations:
- An association with the Task to which the Subtask belongs
- An optional association with zero or one related Subtask, of which the starting point for the Subtask under consideration is dependent (used for time lining of the overall Task eg preceding subtask in a GANTT-schema)
- An optional association with zero, one or many Subtasks, whose starting points are dependent upon the Subtask under consideration (used for time lining of the overall Task, eg succeeding subtasks in a GANTT-schema)

Subtask special recommendations:
- Where a Subtask is dependent upon, eg maintenance level, it must be distinguished by the assignment of an Applicability_statement to the Subtask (refer to Para 4.21, UoF Applicability Statement)

Note
There can be alternative subtasks within a Task. These are also distinguished by the assignment of Applicability_statement.

4.13.3.8 Subtask_timeline
The Subtask_timeline class is association class which adds information to the timeline association between two Subtasks. The timeline association enables the definition of time dependencies between two Subtasks in a Task.

The Subtask_timeline class defines the event to which relating subtask refers (start or end of the related Subtask). It also defines a possible lag, ie duration from the time the related event occurs and the time when the relating Subtask can be initiated (started).

Note
Subtasks that don’t relate to any other Subtask, via the Subtask_timeline association, are to be regarded as starting at the beginning of the associated Task.

Note
This class supports the creation of a GANTT-schema for a Task.

Subtask_timeline attributes:
- Subtask_timeline_event
- Subtask_timeline_lag

4.13.3.9 Subtask_by_reference
The Subtask_by_reference class is a specialization of Subtask. Subtask_by_reference is to be used wherever the Subtask is defined as a Task in its own right.

Note
Subtask_by_reference is the only mechanism in S3000L that supports reuse of task steps and/or task procedures in between Tasks.
Subtask_by_reference attributes:
- Subtask_identifier (inherited from class Subtask)
- Subtask_role (inherited from class Subtask)

Subtask_by_reference associations:
- An association with the Task to which the Subtask_by_reference belongs (inherited from class Subtask)
- An optional association with zero or one related Subtask, of which the starting point for the Subtask_by_reference under consideration is dependent (inherited from class Subtask)
- An optional association with zero, one or many Subtasks, for which the respective starting point is dependent upon the Subtask_by_reference under consideration (inherited from class Subtask)
- An association with the specific Task being referenced as a Subtask

Subtask_by_reference special recommendations:
- See special recommendations for Subtask above

4.13.3.10 Subtask_by_external_reference
The Subtask_by_external_reference class is a specialization of Subtask. Subtask_by_external_reference is to be used wherever the complete description of the Subtask is being described in an external source, i.e., outside the scope of S3000L and S1000D.

Subtask_by_external_reference attributes:
- Subtask_identifier (inherited from class Subtask)
- Subtask_role (inherited from class Subtask)

Subtask_by_external_reference associations:
- An association with the Task to which the Subtask_by_external_reference belongs (inherited from class Subtask)
- An optional association with zero or one related Subtask, of which the starting point for the Subtask_by_external_reference under consideration is dependent (inherited from class Subtask)
- An optional association with zero, one or many Subtasks, for which the respective starting point is dependent upon the Subtask_by_external_reference under consideration (inherited from class Subtask)
- An optional association with zero, one or many instances of External_document (source) containing the detailed description for the subtask to be performed
- An optional association with zero, one or many instances of S1000D_publication_module (source) containing the detailed description for the subtask to be performed

Subtask_by_external_reference special recommendations:
- Where a Subtask_by_external_reference is dependent upon, e.g., maintenance level, it must be distinguished by the assignment of an Applicability_statement to the Subtask_by_external_reference (refer to Para 4.21, UoF Applicability Statement)
- An instance of Subtask_by_external_reference must be associated with at least one instance of External_document or S1000D_publication_module

4.13.3.11 Subtask_by_definition
The Subtask_by_definition class is a specialization of Subtask. A Subtask_by_definition provides a detailed characterization of the subtask to be included in the overall Task. A Subtask_by_definition cannot be referenced from any other Task.

Subtask_by_definition attributes:
− Subtask_identifier (inherited from class Subtask)
− Subtask_role (inherited from class Subtask)
− Subtask_name
− Information_code (zero or one)
− Subtask_description (zero or one)
− Subtask_duration (one or many)
− Maintenance_location_code (zero or one)

Subtask_by_definition associations:
− An association with the Task to which the Subtask_by_definition belongs (inherited from class Subtask)
− An optional association with zero or one related Subtask, of which the starting point for the Subtask_by_definition under consideration is dependent (inherited from class Subtask)
− An optional association with zero, one or many Subtasks, for which the respective starting point is dependent upon the Subtask_by_definition under consideration (inherited from class Subtask)
− An optional association with a zonal location (Zone_element_revision) in which the subtask is to be performed (via Subtask_in_zone)
− An optional association with zero, one or many Subtask_acceptance_parameters, defining criteria that need to be fulfilled before the subtask can be closed
− An optional association with zero, one or many Breakdown_elements and/or parts which are the objects on which the subtask is being performed
− An optional association with zero, one or many Subtask_objective_states that defines the states that must be obtained before the subtask can be closed
− An optional association with zero, one or many Warning_caution_or_notes (via Subtask_warning_caution_or_note)

Subtask_by_definition special recommendations:
− The value for the Subtask_duration attribute can use either of the following representations:
  − Value_with_unit_property
  − Value_with_limit_property, where Limit_qualifier is set to "maximum".
  − Value_with_tolerances_property
  − Value_range_property
− Where the Subtask_durations depends on, eg maintenance level, these are represented as different values (instances of Prp) for the attribute but must be distinguished by assigning Applicability_statement to the respective Prp instance representing the individual value
− See also special recommendations for Subtask above

4.13.3.12 Subtask_in_zone
The Subtask_in_zone class defines a relationship between an instance of Subtask_by_definition and a zonal location (Zone_element_revision) in which the subtask is to be performed.

Note
There is one instance of Subtask_in_zone per relevant combination of Subtask_by_definition and zonal location (Zone_element_revision).

Subtask_in_zone associations:
− An association with the specific Subtask_by_definition that is to be performed
− An association with the Zone_element_revision that represents the zonal location in which the subtask is to be performed
4.13.3.13 Subtask_acceptance_parameter
The Subtask_acceptance_parameter class defines criteria’s that needs to be fulfilled before the subtask can be closed.

Note
An instance of class Subtask_acceptance_parameter has no meaning by itself, but only in the context of a specific instance of Subtask_by_definition.

Subtask_acceptance_parameter attributes:
- Subtask_acceptance_parameter_description
- Subtask_acceptance_parameter_value (one or many)

Subtask_acceptance_parameter associations:
- An association with the specific instance of Subtask_by_definition for which the Subtask_acceptance_parameter has been defined

Subtask_acceptance_parameter special recommendations:
- The value for the Subtask_acceptance_parameter_value attribute can use either of the following representations:
  - Value_with_unit_property
  - Value_with_limit_property, where Limit_qualifier is set to ‘maximum’, or ‘minimum’.
  - Value_with_tolerances_property
  - Value_range_property
- Where the Subtask_acceptance_parameter depends on, eg operational environment, these are represented as different instances of Subtask_acceptance_parameter. The respective instance must be distinguished by the assignment of an Applicability_statement to the respective instance of Subtask_acceptance_parameter (also refer to Para 4.21, UoF Applicability Statement).

4.13.3.14 Subtask_target_item
The Subtask_target_item interface is implemented by classes that can represent the object on which a Subtask_by_definition is to be performed.

Classes that implement the Subtask_target_item interface:
- Breakdown_element_revision
- Part

Classes that implement the Subtask_target_item interface must implement the following association:
- An optional association with zero, one or many instances of Subtask_by_definition

4.13.3.15 Subtask_objective_state
The Subtask_objective_state class defines a state that must be obtained before the subtask can be closed.

Note
An instance of class Subtask_objective_state has no meaning by itself, but only in the context of a specific instance of Subtask_by_definition.

Subtask_objective_state attributes:
- Subtask_objective_state

Subtask_objective_state association:
- An association with the specific instance of Subtask_by_definition for which the Subtask_objective_state has been defined

**Note**

If a Subtask_by_definition must obtain more than one state, there needs to be one instance of Subtask_objective_state per state.

### 4.13.3.16 Subtask_circuit_breaker_state

The Subtask_circuit_breaker_state class is a specialization of Subtask_objective_state. The Subtask_circuit_breaker_state class defines a circuit breaker state that must be obtained before the subtask can be closed. The main difference between the generic Subtask_objective_state and Subtask_circuit_breaker_state is that the Subtask_circuit_breaker_state identifies the individual Circuit_breaker that must obtain the state.

**Note**

An instance of class Subtask_circuit_breaker_state has no meaning by itself, but only in the context of a specific instance of Subtask_by_definition.

**Subtask_circuit_breaker_state attributes:**
- Subtask_objective_state (inherited from class Subtask_objective_state)

**Subtask_circuit_breaker_state associations:**
- An association with the specific instance of Subtask_by_definition for which the Subtask_circuit_breaker_state has been defined (inherited from class Subtask_objective_state)
- An association with the individual Circuit_breaker that must obtain the state

**Note**

If a Subtask_by_definition must obtain more than one circuit breaker state, there needs to be one instance of Subtask_circuit_breaker_state per circuit breaker.

### 4.13.3.17 Circuit_breaker

The Circuit_breaker class identifies individual circuit breakers that are installed on the Product.

**Circuit_breaker attributes:**
- Circuit_breaker_identifier
- Circuit_breaker_name
- Circuit_breaker_type

**Circuit_breaker associations:**
- An optional association with zero, one or many Subtask_circuit_breaker_states, ie states that the circuit breaker must be in before a specific instance of Subtask_by_definition can be closed

**Circuit_breaker class must also implements the following interface:**
- Document_assignment_interface.

**Circuit_breaker special recommendations:**
- Circuit breaker location can be defined using the assignment of a Document, where the Document_assignment_role is set to "Source", and the Document_portion attribute describes the location.

### 4.13.3.18 Warning_caution_or_note

The Warning_caution_or_note class defines advices concerning safety, legal and health aspects.
Warning_caution_or_note attributes:
- Warning_caution_or_note_identifier (zero or one)
- Warning_caution_or_note_description
- Warning_caution_or_note_type

Warning_caution_or_note associations:
- An optional association with zero, one or many Tasks for which the Warning_caution_or_note is valid (via Task_warning_caution_or_note)
- An optional association with zero, one or many specific instances of Subtask_by_definition for which the Warning_caution_or_note is valid (via Subtask_warning_caution_or_note)

4.13.3.19 Task_warning_caution_or_note
The Task_warning_caution_or_note class defines a relationship between a specific Task and an instance of Warning_caution_or_note.

Task_warning_caution_or_note associations:
- An association with the specific Task that needs the advice concerning safety, legal and health aspects
- An association with the instance of Warning_caution_or_note that contains the advice concerning safety, legal and health aspects

Note
There is one instance of Task_warning_caution_or_note per relevant combination of Task and Warning_caution_or_note.

Task_warning_caution_or_note special recommendations:
- Where a Task_warning_caution_or_note is dependent upon eg maintenance level, or external conditions, it must be distinguished by the assignment of an Applicability_statement (UoF Applicability Statement) to the Task_warning_caution_or_note instance.

4.13.3.20 Subtask_warning_caution_or_note
The Subtask_warning_caution_or_note class defines a relationship between specific instance of Subtask_by_definition and an instance of Warning_caution_or_note.

Subtask_warning_caution_or_note associations:
- An association with the specific instance of Subtask_by_definition that needs the advice concerning safety, legal and health aspects
- An association to the instance of Warning_caution_or_note that contains the advice concerning safety, legal and health aspects

Note
There is one instance of Subtask_warning_caution_or_note per relevant combination of Subtask_by_definition and Warning_caution_or_note.

Subtask_warning_caution_or_note special recommendations:
- Where a Subtask_warning_caution_or_note is dependent upon eg maintenance level, or external conditions, it must be distinguished by the assignment of an Applicability_statement (UoF Applicability Statement) to the Subtask_warning_caution_or_note instance.

4.13.4 UoF Task - Additions to referenced class and interface definitions
4.13.4.1 Task_requirement
The Task_requirement class defined in UoF LSA Candidate Task Requirement, must also implement the following additional association:
An optional association with zero, one or many Tasks which are justified by the Task_requirement.

4.13.4.2 S1000D_data_module
The S1000D_data_module class defined in UoF Document must also implement the following association:

− An optional association with zero, one or many Tasks (ie revisions thereof), which are described in the identified issue of the S1000D data module

4.13.4.3 Part
The Part class defined in UoF Part, must also implement the following additional interface:

− Subtask_target_item

4.13.4.4 Breakdown_element_revision
The Breakdown_element_revision class defined in UoF Breakdown Structure, must also implement the following additional interface:

− Subtask_target_item

4.13.4.5 External_document
The External_document class defined in UoF Document, must also implement the following additional association:

− An optional association with zero, one or many instances of Subtask_by_external_reference

4.13.4.6 S1000D_publication_module
The S1000D_publication_module class defined in UoF Document, must also implement the following additional association:

− An optional association with zero, one or many instances of Subtask_by_external_reference

4.13.4.7 Zone_element_revision
The Zone_element_revision class defined in UoF Breakdown Zone Element, must also implement the following additional association:

− An optional association with zero, one or many specific instances of Subtask_by_definition (via Subtask_in_zone)

4.13.4.8 Organization
The Organization class defined in UoF Project must also implement the following additional association:

− An optional association with zero, one or many instances of Task in the role of responsible organization (task origin).

4.14 UoF Task Resources
4.14.1 Overall description
The Task Resources UoF supports a detailed definition of resources needed, either per subtask, or aggregated per task.

Task Resources can be either:

− Material resources, eg spares, tools, consumables
− Facilities, eg hangars, power
− Personnel resources, eg skills
− Documents

Material and facility resources can be defined both in terms of specifications, and in terms of actual parts.
4.14.3 UoF Task Resources - New class and interface definitions

4.14.3.1 Task_resource_assignment

The Task_resource_assignment interface enables resources to be assigned to classes that implement this interface.
Classes that implement the Task_resource_assignment interface:

- Task
- Subtask_by_definition

Classes that implement the Task_resource_assignment interface must also implement the following association:

- An optional association with zero, one or many Task_resource, i.e. resources needed for performing the specific instance of Task or Subtask_by_definition

4.14.3.2 Task_resource

The Task_resource class supports the identification of resources needed for the performance of a specific instance of Task or Subtask_by_definition.

**Note**

Task_resource is an abstract class, i.e. an instantiation of Task_resource must be either:

- Task_material_resource_by_specification
- Task_materiel_resource_by_reference
- Task_facility_resource_by_specification
- Task_facility_resource_by_reference
- Task_personnel_resource
- Task_document.

**Note**

Each instance of Task_resource is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse of resource definitions is enabled through the use of Resource_specifications, Parts, Skills, Trades and Documents.

Task_resource attributes:

- Fixed_resource_marker
- Task_resource_duration

Task_resource associations:

- An association with a specific instance of Task or Subtask_by_definition, for which the associated resource is needed (via Task_resource_assignment interface)
- An optional association where an instance of Task_resource can relate to other instances of Task_resource, e.g. “person A uses tool B”
- An optional association where an instance of Task_resource can be related to from other instances of Task_resource, e.g. “tool B is used by person A”

Task_resource special recommendations:

- Where a Task_resource is dependent upon e.g., maintenance level, external conditions, etc., it must be distinguished by the assignment of an Applicability_statement (UoF Applicability Statement) to the Task_resource instance
- Where a Task_resource_duration is depending on e.g., maintenance level, external conditions etc., these must be represented as a set of values (Prp instances), but be distinguished by the assignment of Applicability_statement to the respective Prp instance that represents the individual values.

4.14.3.3 Task_resource_relationship

The Task_resource_relationship class is an association class which adds information to the Task_resource_relationship association between two instances of Task_resource. The Task_resource_relationship association defines dependencies between two Task_resources in a Task or in Subtask_by_definition, e.g. person A “uses” tool B.
Task_resource_relationship attributes:

- Task_resource_relationship_category

4.14.3.4 Task_material_resource

The Task_material_resource class is a specialization of the Task_resource class, and identifies material resources needed for the performance of a specific instance of Task or Subtask_by_definition.

**Note**

Task_material_resource is an abstract class, ie an instantiation of Task_material_resource must be either:

- Task_material_resource_by_specification
- Task_material_resource_by_reference

**Note**

Each instance of Task_material_resource is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse of material resource definitions is enabled through the use of Resource_specifications and Parts.

Task_material_resource attributes:

- Fixed_resource_marker (inherited from class Task_resource)
- Task_resource_duration (inherited from class Task_resource)
- Task_material_resource_quantity
- Task_material_resource_category (zero or one)

Task_material_resource associations:

- An association with a specific instance of Task or Subtask_by_definition for which the associated resource is needed (inherited from class Task_resource)
- An optional association where an instance of Task_material_resource can relate to other instances of Task_resource, eg “equipment A calibrates tool B” (inherited from class Task_resource)
- An optional association where an instance of Task_material_resource can be related to from other instances of Task_resource, eg “tool B is calibrated by equipment A”. (inherited from class Task_resource)

Task_material_resource special recommendations:

- See Task_resource special recommendations

**Note**

For a Task/Subtask_by_definition it’s recommended to define every item that is affected by the task/subtask as a Task material resource. The consumption rate then can be calculated based on the scrap rate defined for the Hardware_part, eg if four screws are removed in order to open a hatch, these will be recorded as task material resources, and the consumption will be calculated based on Task_material_resource_quantity and Hardware_part_scrap_rate.

4.14.3.5 Task_material_resource_by_reference

The Task_material_resource_by_reference class is a specialization of the Task_material_resource class, and identifies a specific Part (by Part_identifier) as the resource needed for the performance of a specific Task or Subtask_by_definition.

**Note**

Each instance of Task_material_resource_by_reference is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse is enabled through the use of Parts.
Task_material_resource_by_reference attributes:
- Fixed_resource_marker (inherited from class Task_resource)
- Task_resource_duration (inherited from class Task_resource)
- Task_material_resource_quantity (inherited from class Task_material_resource)
- Task_material_resource_category (inherited from class Task_material_resource)

Task_material_resource_by_reference associations:
- An association with a specific instance of Task or Subtask_by_definition for which the associated resource is needed (inherited from class Task_resource)
- An optional association where an instance of Task_material_resource_by_reference can relate to other instances of Task_resource, eg “equipment A calibrates tool B” (inherited from class Task_resource)
- An optional association where an instance of Task_material_resource_by_reference can be related to from other instances of Task_resource, eg “tool B is calibrated by person A” (inherited from class Task_resource)
- An association with the Part that plays the role of a Task_material_resource

Task_material_resource_by_reference special recommendations:
- See Task_resource special recommendations

Note
- For a Task/Subtask_by_definition it’s recommended to define every item that is affected by the task/subtask as a Task material resource. The consumption rate then can be calculated based on the scrap rate defined for the Hardware_part, eg if four screws are removed in order to open a hatch, these will be recorded as task material resources, and the consumption will be calculated based on Task_material_resource_quantity and Hardware_part_scrap_rate.

4.14.3.6 Task_material_resource_by_specification
The Task_material_resource_by_specification class is a specialization of the Task_material_resource class, and identifies material resources needed to perform a specific Task or Subtask_by_definition, through a Resource_specification.

Note
- Each instance of Task_material_resource_by_specification is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse of material resource specifications is enabled through the use of Resource_specification.

Task_material_resource_by_specification attributes:
- Fixed_resource_marker (inherited from class Task_resource)
- Task_resource_duration (inherited from class Task_resource)
- Task_material_resource_quantity (inherited from class Task_material_resource)
- Task_material_resource_category (inherited from class Task_material_resource)

Task_material_resource_by_specification associations:
- An association with a specific instance of Task or Subtask_by_definition, for which the associated resource is needed (inherited from class Task_resource)
- An optional association where an instance of Task_material_resource_by_specification can relate to other instances of Task_resource, eg “equipment A calibrates tool B” (inherited from class Task_resource)
- An optional association where an instance of Task_material_resource_by_specification can be related to from other instances of Task_resource, eg “tool B is calibrated by person A” (inherited from class Task_resource)
An association with an instance of Material_resource_specification that plays the role of a Task_material_resource

Task_material_resource_by_specification special recommendations:

− See Task_resource special recommendations

Note

For a Task/Subtask_by_definition it’s recommended to define every item that is affected by the task/subtask as a Task material resource. The consumption rate then can be calculated based on the scrap rate defined for the Hardware_part, e.g., if four screws are removed in order to open a hatch, these will be recorded as task material resources, and the consumption will be calculated based on Task_material_resource_quantity and Hardware_part_scrap_rate.

4.14.3.7 Resource_specification

The Resource_specification class identifies specifications of resources which can be realized by one or many Parts.

Note

Use of Resource_specifications allows for more generic task/subtask definitions, i.e., the task/subtask does not need to be changed based on customer specific tool sets etc.

Note

Resource_specification is an abstract class, i.e., an instantiation of Resource_specification must be either a Facility_resource_specification or a Material_resource_specification.

Resource_specification attributes:

− Resource_specification_identifier
− Resource_specification_name
− Resource_specification_description

Resource_specification associations:

− An optional association with zero, one or many Parts that realizes the Resource_specification, i.e., actual tools, equipment, spares, that can be used when the task is to be performed (via an instance of the Resource_realization class)

4.14.3.8 Resource_realization

The Resource_realization class defines a relationship between a specific Resource_specification and a Part that fulfills that specification.

Resource_realization associations:

− An association with a specific Resource_specification which has a Part realization
− An association with a specific Part that fulfills the resource specification (realizes the specification)

Note

There is one instance of Resource_realization per relevant combination of Resource_specification and Part.

Resource_realization special recommendations:

− Where a Resource_realization is dependent upon e.g., customer, maintenance level or external conditions, it must be distinguished by the assignment of an Applicability_statement to the Resource_realization instance (UoF Applicability Statement)
4.14.3.9 Material_resource_specification

The Material_resource_specification class is a specialization of the Resource_specification class, and identifies specifications of material resources (spares, consumables, tools, etc.) which can be realized by one or many Parts.

Note

Use of Material_resourceSpecifications allows for more generic task/subtask definitions, i.e. the task/subtask does not need to be changed based on customer specific tool sets.

Material_resource_specification attributes:

- Resource_specification_identifier (inherited from class Resource_specification)
- Resource_specification_name (inherited from class Resource_specification)
- Resource_specification_description (inherited from class Resource_specification)

Material_resource_specification associations:

- An optional association with zero, one or many Parts that realizes the Material_resource_specification, i.e. actual tools, equipments, spares, that can be used when the task is to be performed (inherited from class Resource_specification)
- An optional association with zero, one or many instances of Task_material_resource_by_specification, i.e. individual usages of the specified material resource in a specific Task or Subtask_by_definition

Material_resource_specification special recommendations:

- See Resource_specification special recommendations.

4.14.3.10 Task_facility_resource

The Task_facility_resource class is a specialization of the Task_resource class, and identifies facility resources needed for the performance of a specific instance of Task or Subtask_by_definition.

Note

Task_facility_resource is an abstract class, i.e. an instantiation of Task_facility_resource must be either:

- Task_facility_resource_by_specification
- Task_facility_resource_by_reference

Note

Each instance of Task_facility_resource is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse of facility resource definitions is enabled through the use of Resource_specifications and Parts.

Task_facility_resource attributes:

- Fixed_resource_marker (inherited from class Task_resource)
- Task_resource_duration (inherited from class Task_resource)
- Task_facility_resource_quantity (zero or one)

Task_facility_resource associations:

- An association with a specific instance of Task or Subtask_by_definition for which the associated resource is needed (inherited from class Task_resource)
- An optional association where an instance of Task_facility_resource can relate to other instances of Task_resource, e.g. “power utility A powers equipment B” (inherited from class Task_resource)
An optional association where an instance of Task_facility_resource can be related to from other instances of Task_resource, eg "equipment B is powered by power utility A" (inherited from class Task_resource)

Task_material_resource special recommendations:

- See Task_resource special recommendations

4.14.3.11 Task_facility_resource_by_reference

The Task_facility_resource_by_reference class is a specialization of the Task_facility_resource class, and identifies a specific Part (by Part_identifier) as the resource needed for the performance of a specific Task or Subtask_by_definition.

**Note**

Each instance of Task_facility_resource_by_reference is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse is enabled through the use of Parts.

Task_facility_resource_by_reference attributes:

- Fixed_resource_marker (inherited from class Task_resource)
- Task_resource_duration (inherited from class Task_resource)
- Task_facility_resource_quantity (zero or one, inherited from class Task_facility_resource)

Task_facility_resource_by_reference associations:

- An association with a specific instance of Task or Subtask_by_definition for which the associated resource is needed (inherited from class Task_resource)
- An optional association where an instance of Task_facility_resource_by_reference can relate to other instances of Task_resource, eg "power utility A powers equipment B" (inherited from class Task_resource)
- An optional association where an instance of Task_facility_resource_by_reference can be related to from other instances of Task_resource, eg "equipment B is powered by power utility A" (inherited from class Task_resource)
- An association with the Part that plays the role of a Task_facility_resource

Task_facility_resource_by_reference special recommendations:

- See Task_resource special recommendations

4.14.3.12 Task_facility_resource_by_specification

The Task_facility_resource_by_specification class is a specialization of the Task_facility_resource class, and identifies facility resource needed to perform a specific Task or Subtask_by_definition through a Resource_specification.

**Note**

Each instance of Task_facility_resource_by_specification is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse of facility resource specifications is enabled through the use of Resource_specification.

Task_facility_resource_by_specification attributes:

- Fixed_resource_marker (inherited from class Task_resource)
- Task_resource_duration (inherited from class Task_resource)
- Task_facility_resource_quantity (inherited from class Task_facility_resource)

Task_facility_resource_by_specification associations:

- An association with a specific instance of Task or Subtask_by_definition for which the associated resource is needed (inherited from class Task_resource)
An optional association where an instance of Task_facility_resource_by_specification can relate to other instances of Task_resource, eg “power utility A powers equipment B” (inherited from class Task_resource)

An optional association where an instance of Task_facility_resource_by_specification can be related to from other instances of Task_resource, eg “equipment B is powered by power utility A” (inherited from class Task_resource)

An association with an instance of Facility_resource_specification that plays the role of a Task_facility_resource

Task_facility_resource_by_specification special recommendations:
- See Task_resource special recommendations

4.14.3.13 Facility_resource_specification
The Facility_resource_specification class is a specialization of the Resource_specification class, and identifies specifications of facility resources (power, internet, hangar, dock etc) which can be realized by one or many Parts.

Note
Use of Facility_resource_specifications allows for more generic task/subtask definitions, ie the task/subtask does not need to be changed based on customer specific realizations of facilities etc.

Facility_resource_specification attributes:
- Resource_specification_identifier (inherited from class Resource_specification)
- Resource_specification_name (inherited from class Resource_specification)
- Resource_specification_description (inherited from class Resource_specification)

Facility_resource_specification associations:
- An optional association with zero, one or many Parts that realizes the Facility_resource_specification, ie actual equipment etc, that can be used when the task is to be performed (inherited from class Resource_specification)
- An optional association with zero, one or many Task_facility_resource_by_specifications, ie individual usages of the specified facility resource in a specific Task or Subtask_by_definition

Facility_resource_specification special recommendations:
- See Resource_specification special recommendations.

4.14.3.14 Task_personnel_resource
The Task_personnel_resource class is a specialization of the Task_resource class, and identifies personnel resources needed for the performance of a specific instance of Task or Subtask_by_definition.

Note
Each instance of Task_personnel_resource is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse of personnel resource definitions is enabled through the use of Trades and Skills.

Task_personnel_resource attributes:
- Fixed_resource_marker (inherited from class Task_resource)
- Task_resource_duration (inherited from class Task_resource)
- Task_personnel_resource_role
- Task_number_of_personnel_resource (zero, one or many)
- Task_personnel_resource_labour_time (zero, one or many)
Task_personnel_resource associations:
- An association with a specific instance of Task or Subtask_by_definition for which the associated resource is needed (inherited from class Task_resource).
- An optional association where an instance of Task_personnel_resource can relate to other instances of Task_resource, eg "person A uses equipment B" (inherited from class Task_resource).
- An optional association where an instance of Task_personnel_resource can be related to other instances of Task_resource, eg "equipment B is used by person A" (inherited from class Task_resource).
- An optional association with zero, one or many Competences (Trades or Skills) that can perform as the needed Task_personnel_resource (via Task_personnel_resource_competence).

Task_material_resource special recommendations:
- Where a Task_personnel_resource is dependent upon eg maintenance level, external conditions, etc, it must be distinguished by assigning an Applicability_statement (UoF Applicability) to the Task_personnel_resource instance.
- Where a Task_personnel_resource has different Task_resource_duration, Task_number_of_personnel_resource, and/or Task_personnel_resource_labour_time depending on eg maintenance level, these must be represented as a set of values (Prp instances) for the respective attribute, but be distinguished by the assignment of Applicability_statement to the respective Prp instance representing the individual values.

4.14.3.15 Competence
The Competence interface enables Skills and Trades to be associated with personnel resource requirements, needed to perform a Task or a Subtask_by_definition.

Classes that implement the Competence interface:
- Trade
- Skill

Classes that implement the Competence interface must also implement the following association:
- An optional association with zero, one or many instances of Task_personnel_resource, ie personnel needed for the performance of a specific instance of Task or Subtask_by_definition (via Task_personnel_resource_competence).

4.14.3.16 Trade
The Trade class defines types of occupations.

Trade attributes:
- Trade_name

Trade associations:
- An optional association with zero, one or many Skills defined for the Trade

Trade interfaces:
- The Trade class implements the Competence interface

4.14.3.17 Skill
The Skill class defines specific combinations of Trade and Skill_level.

Skill attributes:
- **Skill_code**

  **Skill associations:**
  - An optional association with zero or one Trade for which the Skill is defined

  **Trade interfaces:**
  - The Trade class implements the Competence interface

  **4.14.3.18 Skill_level**
  The Skill_level class is an association class which adds information to the association between a Skill and a Trade.

  **Skill_level attributes:**
  - **Skill_level_name** (zero or one)

  **4.14.3.19 Task_personnel_resource_competence**
  The Task_personnel_resource_competence class defines a relationship between an instance of Task_personnel_resource (ie a needed type of personnel) and a specific Competence (ie an instance of Skill or Trade) that fulfills that need.

  **Task_personnel_resource_competence associations:**
  - An association with an instance of Task_personnel_resource which defines the need of a personnel resource for a specific instance of Task or Subtask_by_definition
  - An association with a specific Competence that has the required qualifications for performing the defined task
  - An optional association with zero, one or many Special_training_requirements which are defined in order for the defined Competence to be able to perform the defined task

  **Note**
  There is one instance of Task_personnel_resource_competence per relevant combination of Task_personnel_resource and Competence (ie Trade or Skill).

  **Task_personnel_resource_competence special recommendations:**
  - Where a specified Task_personnel_resource_competence is dependent upon eg customer, maintenance level, or external conditions, it must be distinguished by the assignment of an Applicability_statement to the Task_personnel_resource_competence instance (refer to Para 4.21, UoF Applicability statement)

  **4.14.3.20 Special_training_requirement**
  The Special_training_requirement class identifies additional training required for the given Competence, in order for the defined Competence to be qualified to perform the defined task.

  **Note**
  An instance of class Special_training_requirement has no meaning by itself, but only in the context of an association between a specific required personnel resource, and a defined Competence (ie in association with an instance of Task_personnel_resource_competence).

  **Special_training_requirement attributes:**
  - **Special_training_requirement_description**
  - **Training_method** (zero or one)

  **Special_training_requirement associations:**
  - An instance of Special_training_requirement is always associated with a specific instance of Task_personnel_resource_competence (ie the association between a needed personnel resource and a specific Competence that fulfills the need for a personnel resource).
4.14.3.21 Task_document
The Task_document class is a specialization of the Task_resource class, and identifies Documents needed for the performance of a specific Task or Subtask_by_definition.

Note
Each instance of Task_document is uniquely defined for a specific instance of Task or Subtask_by_definition. Reuse of document resource definitions is enabled through the use of Documents.

Task_document attributes:
- Fixed_resource_marker (inherited from class Task_resource)
- Task_resource_duration (inherited from class Task_resource)

Task_document associations:
- An association with a specific instance of Task or Subtask_by_definition for which the associated Task_document resource is needed (inherited from class Task_resource)
- An optional association where an instance of Task_document can relate to other instances of Task_resource (inherited from class Task_resource)
- An optional association where an instance of Task_document can be related to from other instances of Task_resource (inherited from class Task_resource)
- An association with the specific Document that is to be used in the role of a task resource

Task_material_resource special recommendations:
- See Task_resource special recommendations

4.14.4 UoF Task Resources - Additions to referenced class and interface definitions
4.14.4.1 Task
The Task class defined in UoF Task, must also implement the following additional interface:
- Task_resource_assignment

4.14.4.2 Subtask_by_definition
The Subtask_by_definition class defined in UoF task, must also implement the following additional interface:
- Task_resource_assignment

4.14.4.3 Document
The Document class defined in UoF document, must also implement the following additional association:
- An optional association with zero, one or many instances of Task_document, ie situations where the Document is to be used as resource in a task

4.14.4.4 Part
The Part class defined in UoF part, must also implement the following additional associations:
- An optional association with zero, one or many instances of Resource_realization, ie areas where the Part meets the requirements in a Resource_specification, and hence can be used as a resource in one or many tasks
- An optional association with zero, one or many instances of Task_material_resource_by_reference, ie instances where the Part has been identified as the material resource to be used in a task
An optional association with zero, one or many instances of Task_facility_resource_by_reference, ie instances where the Part has been identified as the facility resource to be used in a task

4.15 UoF Task Usage (Part 1)
4.15.1 Overall description
The Task Usage (Part 1) UoF supports detailed definitions for:

- Identifying the breakdown element or part on which an operational or rectifying task is to be performed
- Thresholds and triggers defining the conditions that initiates an operational or rectifying task
- Where the operational or rectifying task is to be performed
4.15.2 Graphical representation

Fig 30 UoF Task Usage (Part 1) - class model
4.15.3 UoF Task Usage (Part 1) - New class and interface definitions

4.15.3.1 Task_target_item
The Task_target_item interface is implemented by classes on which a Task can be performed.

Classes that implement the Task_target_item interface:
- Breakdown_element_revision
- Part

Classes that implement the Task_target_item interface must implement the following associations:
- An optional association with zero, one or many instances of Task_usage

Note
Since the Task_target_item interface is implemented by Breakdown_element_revision and Part, it means that instances of the following classes can be selected as Task_target_items: Hardware_element_revision, Software_element_revision, Aggregated_element_revision, Zone_element_revision, Hardware_part and Software_part respectively.

4.15.3.2 Task_usage
The Task_usage class (specifically its subclasses) identifies the Task that is to be performed on a Breakdown_element_revision or Part (ie the Task_target_item).

Note
Task_usage is an abstract class, ie an instantiation of Task_usage needs to be either a Planned_task_usage or Supporting_task_usage (refer to Para 4.16, UoF Task Usage (Part 2)).

Task_usage associations:
- An association with the Task_target_item (instance of Breakdown_element_revision or Part) on which the associated task is to be performed
- An optional association with zero, one or many instances of Task_frequency

4.15.3.3 Task_frequency
The Task_frequency class defines the frequency of performance or occurrence for the associated Task.

Note
An instance of class Task_frequency has no meaning by itself, but only in the context of a specific Task_usage.

Task_frequency attributes:
- Task_frequency
- Task_frequency_calculation_method (zero or one)

Task_frequency associations:
- An instance of Task_frequency is always associated with a specific instance of Task_usage, ie a specific combination of Task and the Task_target_item on which the task is to be performed.

Task_frequency special recommendations:
- Where a Task_frequency is dependent upon eg external conditions, customer, etc, it must be distinguished by the assignment of an Applicability_statement to the Task_frequency instance (also refer to Para 4.21, UoF Applicability Statement).
4.15.3.4 Planned_task_item
The Planned_task_item interface is implemented by classes that can be associated with a Planned_task_usage, i.e., be associated with task limits and preferred locations for task execution.

Classes that implement the Planned_task_item interface:
- Rectifying_task
- Operational_task

Classes that implement the Planned_task_item interface must also implement the following associations:
- An association with one or many instances of Planned_task_usage

4.15.3.5 Planned_task_usage
The Planned_task_usage class is a specialization of the Task_usage class, and adds information in terms of task limits and preferred locations for Task execution.

**Note**
There is one instance of Planned_task_usage per relevant combination of Planned_task_item and Breakdown_element_revision or Part.

**Note**
The Planned_task_usage class must be used to associate a Rectifying_task or an Operational_task with the item on which the Task is to be performed.

Planned_task_usage associations:
- An association with the item (instance of Breakdown_element_revision or Part via the Task_target_item interface) on which the associated Planned_task_item is to be performed (inherited from class Task_usage).
- An optional association with zero, one or many instances of Task_frequency (inherited from class Task_usage).
- An association with the Planned_task_item (instance of Rectifying_task or Operational_task) which is to be performed on the Task_target_item (instance of Breakdown_element_revision or Part).
- An association with at least one instance of Maintenance_level_allocation, i.e., location where the associated Planned_task_item is to be performed.
- An optional association with zero, one or many instances of Task_limit, i.e., thresholds (intervals) defining when the associated Planned_task_item is to be performed.

Planned_task_usage recommendations:
- Where an instance of Planned_task_usage is dependent upon e.g., external conditions, customer, etc., it must be distinguished by the assignment of an Applicability_statement (also refer to Para 4.21, UoF Applicability Statement).

**Note**
Associating a Planned_task_usage with a specific maintenance level, by default means that any maintenance level above can carry out the same Task.

4.15.3.6 Maintenance_level_allocation
The Maintenance_level_allocation class defines a relationship between an instance of Planned_task_usage and an identified support organization where the associated Planned_task_item is to be carried out.

**Note**
An instance of class Maintenance_level_allocation has no meaning by itself, but only in the context of a Planned_task_usage.
Maintenance_level_allocation associations:

- An instance of Maintenance_level_allocation is always associated with an instance of Planned_task_usage, i.e., specific combination of a Rectifying_task or Operational_task and Breakdown_element_revision or Part.
- An association with the Maintenance_level_type, Maintenance_location, Operating_location_type, or Operating_location that is identified as the organization that must have the capability to perform the associated task (via the Maintenance_level_allocation_interface).

**Note**

There is one instance of Maintenance_level_allocation per relevant combination of Planned_task_usage and support organization.

Maintenance_level_allocation special recommendations:

- Where a Maintenance_level_allocation is dependent upon e.g., operational scenario, etc., it must be distinguished by the assignment of an Applicability_statement to the instance of Maintenance_level_allocation (also refer to Para 4.21, UoF Applicability Statement).

4.15.3.7 Maintenance_level_allocation_interface

The Maintenance_level_allocation_interface is implemented by classes that can represent support organizations where a Planned_task_item can be performed.

Classes that implement the Maintenance_level_allocation_interface interface:

- Maintenance_level_type
- Maintenance_location
- Operating_location_type
- Operating_location

Classes that implement the Maintenance_level_allocation_interface must also implement the following association:

- An optional association with zero, one, or many instances of Maintenance_level_allocation

4.15.3.8 Task_limit

The Task_limit class identifies thresholds (intervals) which defines when the associated Planned_task_item is to be performed on the identified Breakdown_element_revision or Part.

**Note**

An instance of class Task_limit has no meaning by itself, but only in the context of a specific Planned_task_usage or Task_requirement_target (UoF LSA Candidate Task Requirement).

**Note**

Task_limit is an abstract class, i.e., an instantiation of Task_limit needs to be either a Discrete_task_limit or a Periodic_task_limit.

Task_limit attributes:

- Task_limit_harmonization_indicator
- Task_limit_description (zero or one)

Task_limit associations:

- An instance of Task_limit is always associated with a specific instance of Planned_task_usage or Task_requirement (Refer to UoF LSA Candidate Task Requirement)

Task_limit special recommendations:
Where an instance of Task_limit is dependent upon eg external conditions, customer, etc, it must be distinguished by the assignment of an Applicability_statement (also refer to Para 4.21 UoF Applicability Statement)

4.15.3.9 Sampling
The Sampling class identifies the method to be used for selecting a limited set of manufactured products, on which the task is to be performed.

Note
An instance of class Sampling has no meaning by itself, but only in the context of a specific Discrete_task_limit, Initial_task_limit or Repeat_task_limit.

Note
An instance of Sampling can be (but must not be) specialized into either Sampling_by_ratio or Sampling_by_value.

Sampling attributes:
- Sampling_method_description

Sampling associations:
- An instance of Sampling must always be associated with a specific instance of a class that implements the Sampling_interface.

4.15.3.10 Sampling_by_ratio
The Sampling_by_ratio class is a specialization of the Sampling class, and identifies that the method used for selecting products on which the rectifying task is to be performed is done by ratio.

Note
An instance of class Sampling_by_ratio has no meaning by itself, but only in the context of a specific Discrete_task_limit, Initial_task_limit or Repeat_task_limit.

Sampling_by_ratio attributes:
- Sampling_method_description (inherited from class Sampling)
- Sampling_method_ratio

Sampling_by_ratio associations:
- An instance of Sampling_by_ratio must always be associated with a specific instance of a class that implements the Sampling_interface (inherited from class Sampling)

4.15.3.11 Sampling_by_value
The Sampling_by_value class is a specialization of the Sampling class, and identifies that the method used for selecting products on which the rectifying task is to be performed is done by value.

Note
An instance of class Sampling_by_value has no meaning by itself, but only in the context of a specific Discrete_task_limit, Initial_task_limit or Repeat_task_limit.

Sampling_by_value attributes:
- Sampling_method_description (inherited from class Sampling)
- Sampling_method_value

Sampling_by_value associations:
An instance of Sampling_by_value must always be associated with a specific instance of a class that implements the Sampling_interface (inherited from class Sampling).

### 4.15.3.12 Sampling_interface

The Sampling_interface is implemented by classes that can be associated with a defined sampling.

**Classes that implement the Sampling_interface:**

- Discrete_task_limit
- Initial_task_limit
- Repeat_task_limit

**Classes that implement the Sampling_interface must implement the following associations:**

- An optional association with zero or one instance of Sampling (and subclasses thereof)

### 4.15.3.13 Discrete_task_limit

The Discrete_task_limit class is a specialization of the Task_limit class, and is used to identify task limits that are distinct from each other, i.e., there is no scheduled next occurrence for the associated task.

**Note**

An instance of class Discrete_task_limit has no meaning by itself, but only in the context of a specific instance of Planned_task_usage.

**Note**

Discrete_task_limit must not contain a computable expression, i.e., it can just contain a Task_limit_description without any computable triggers or thresholds.

**Discrete_task_limit attributes:**

- Task_limit_harmonization_indicator (inherited from class Task_limit)
- Task_limit_description (inherited from class Task_limit)

**The Discrete_task_limit class must implement the following interfaces:**

- Sampling_interface

**Discrete_task_limit associations:**

- An instance of Discrete_task_limit is always associated with a specific instance of Planned_task_usage or Task_requirement (inherited from class Task_limit)
- An optional trigger association with zero, one or many instances of Threshold_definition
- An optional threshold association with zero, one or many instances of Threshold_definition

**Note**

The threshold association defines a specified interval, from a defined event, to the next required performance of the Planned_task_item. The defined event is identified by the trigger association.

**Note**

The trigger association describes the event to which the threshold is related, e.g., bird strike or engine replacement. If the Discrete_task_limit does not define any trigger association, it is to be assumed that the trigger event equals the Hardware_part_maintenance_start defined for Hardware_part in UoF Part. The trigger activates the threshold.

**Note**

Multiple instances of Threshold_definitions for both the trigger and the threshold association mean ‘whichever comes first’.
Discrete_task_limit special recommendations:

- Where an instance of Discrete_task_limit is dependent upon eg external conditions, customer, etc, it must be distinguished by the assignment of an Applicability_statement (also refer to Para 4.21, UoF Applicability Statement)

Note
The Discrete_task_limit can be used to define "Perform once" and "On condition" task limits as defined in S1000D.

4.15.3.14 Periodic_task_limit

The Periodic_task_limit class is a specialization of the Task_limit class, and is used to identify task limits that are repeated with a specific interval, ie there is a next scheduled occurrence for the associated task.

Note
An instance of class Periodic_task_limit has no meaning by itself, but only in the context of a specific instance of Planned_task_usage plan.

Note
Periodic_task_limit must contain a computable expression.

Periodic_task_limit attributes:

- Task_limit_harmonization_indicator (inherited from class Task_limit)
- Task_limit_description (inherited from class Task_limit)

Periodic_task_limit associations:

- An instance of Periodic_task_limit is always associated with a specific instance of Planned_task_usage or Task_requirement (inherited from class Task_limit)
- An optional initial threshold association with an instance of Initial_task_limit
- An association with one or many instances of Repeat_task_limit

Note
The Initial_task_limit association defines a specified interval for the first required performance of the Planned_task_item (ie Rectifying_task or Operational_task). The interval is to be assumed to be triggered by the condition that is defined as the Hardware_part_maintenance_start for Hardware_part in UoF Part.

Note
Multiple instances of Repeat_task_limits for the repeat association require that each instance of Repeat_task_limits is ordered using the Subsequent_repeat_relationship class.

Periodic_task_limit special recommendations:

- Where an instance of Periodic_task_limit is dependent upon eg external conditions, customer, etc, it must be distinguished by the assignment of an Applicability_statement (also refer to Para 4.21, UoF Applicability Statement)

Note
The Periodic_task_limit class can be used to define explicit combinations of "Perform once" and "Perform periodically" task limits as defined in S1000D.

4.15.3.15 Initial_task_limit

The Initial_task_limit class defines a specific interval for the first scheduled performance of a periodic task.
An instance of class Initial_task_limit has no meaning by itself, but only in the context of a specific Periodic_task_limit.

The Initial_task_limit class must implement the following interfaces:
- Sampling_interface

Initial_task_limit associations:
- An instance of Initial_task_limit is always associated with a specific instance of Periodic_task_limit
- A threshold association with one or many instances of Threshold_definition

Note: The threshold association defines a specified interval for the first performance of the Planned_task_item (ie Rectifying_task or Operational_task).

Note: Multiple instances of Threshold_definition for the threshold association mean “whichever comes first”.

4.15.3.16 Repeat_task_limit
The Repeat_task_limit class defines task limits that are repeated with a specific interval.

Note: An instance of class Repeat_task_limit has no meaning by itself, but only in the context of a specific Periodic_task_limit.

Note: Repeat_task_limit must contain a computable expression.

The Repeat_task_limit class must implement the following interfaces:
- Sampling_interface

Repeat_task_limit associations:
- An instance of Repeat_task_limit is always associated with a specific instance of Periodic_task_limit
- A threshold association with one or many instances of Threshold_definition
- An optional association with zero or one succeeding instance of Repeat_task_limit, via the Subsequent_repeat_relationship
- An optional association with zero or one preceding instance of Repeat_task_limit, via the Subsequent_repeat_relationship

Note: The threshold association defines a specified interval for the repeated performance of the Planned_task_item (ie the associated Rectifying_task or Operational_task).

Note: Multiple instances of Threshold_definition for the threshold association mean ‘whichever comes first’.

Note: The Repeat_task_limit class can be used to define "Perform periodically" task limits as defined in S1000D Issue 3.0.

4.15.3.17 Subsequent_repeat_relationship
The Subsequent_repeat_relationship class defines a subsequent relationship between two instances of Repeat_task_limit.
Note
The definition of when an instance of Repeat_task_limit is to be succeeded by another instance (ie replaced by another instance) is determined by the instance of Threshold_definition defined via the Subsequent_repeat_relationship trigger association.

Subsequent_repeat_relationship associations:
- A preceding association with an instance of Repeat_task_limit
- An succeeding association with an instance of Repeat_task_limit
- A trigger association with one or many instances of Threshold_definition

Note
The trigger association describes the event when the instance of Repeat_task_limit identified by the succeeding association will replace the instance of Repeat_task_limit identified by the preceding association.

Note
Multiple instances of Threshold_definition for the trigger association mean "whichever comes first".

4.15.3.18 Threshold_definition
The Threshold_definition class defines the value or event that constitutes a threshold or trigger.

Note
An instance of class Threshold_definition has no meaning by itself, but only in the context of a specific instance of either:
- Discrete_task_limit, as a trigger
- Discrete_task_limit, as a threshold
- Initial_task_limit, as an initial threshold
- Repeat_task_limit, as a threshold
- Subsequent_repeat_relationship, as a trigger

Note
Threshold_definition is an abstract class, ie an instantiation of Threshold_definition needs to be either a Parameter_threshold or an Event_threshold.

Threshold_definition associations:
- An instance of Threshold_definition can only be associated with one specific instance of one of the following:
  - Discrete_task_limit, as a trigger
  - Discrete_task_limit, as a threshold
  - Initial_task_limit, as an initial threshold
  - Repeat_task_limit, as a threshold
  - Subsequent_repeat_relationship, as a trigger

4.15.3.19 Parameter_threshold
The Parameter_threshold class is a specialization of class Threshold_definition and is used to represent a value that constitutes the threshold or trigger.

Note
An instance of class Parameter_threshold has no meaning by itself, but only in the context of a specific instance of either:
- Discrete_task_limit, as a trigger
- Discrete_task_limit, as a threshold
- Initial_task_limit, as an initial threshold
• Repeat_task_limit, as a threshold
• Subsequent_repeat_relationship, as a trigger.

Note
An example of Parameter_threshold is 1000 flight hours. This value can either be a trigger or a threshold depending on how it is being referenced.

Parameter_threshold attributes:
- Threshold_value

Parameter_threshold associations:
- An instance of Parameter_threshold is always associated with a specific instance of one of the following:
  • Discrete_task_limit, as a trigger
  • Discrete_task_limit, as a threshold
  • Initial_task_limit, as an initial threshold
  • Repeat_task_limit, as a threshold
  • Subsequent_repeat_relationship, as a trigger

Parameter_threshold special recommendations:
- A representation of the Threshold_value attribute is recommended to use the Value_with_limit_property representation, where Limit_qualifier is set to ‘maximum’. There can also be cases where the use of Value_with_tolerances_property or Value_range_property is valid

4.15.3.20 Event_threshold
The Event_threshold class is a specialization of class Threshold_definition and is used to represent the number of occurrences of an explicit event that constitutes the threshold or trigger.

Note
An instance of class Event_threshold has no meaning by itself, but only in the context of a specific instance of either:
• Discrete_task_limit, as a trigger
• Discrete_task_limit, as a threshold
• Initial_task_limit, as an initial threshold
• Repeat_task_limit, as a threshold.
• Subsequent_repeat_relationship, as a trigger

Note
Event_threshold is an abstract class, ie an instantiation of Event_threshold needs to be either, an LSA_failure_mode_related_threshold, a Special_event_related_threshold or a Task_related_threshold.

Event_threshold attributes:
- Event_threshold_number_of_event_occurrences

Event_threshold associations:
- An instance of Event_threshold is always associated with a specific instance of one of the of the following:
  • Discrete_task_limit, as a trigger
  • Discrete_task_limit, as a threshold
  • Initial_task_limit, as an initial threshold
• Repeat_task_limit, as a threshold
• Subsequent_repeat_relationship, as a trigger

4.15.3.21 LSA_failure_mode_related_threshold

The LSA_failure_mode_related_threshold class is a specialization of class Event_threshold. LSA_failure_mode_related_threshold is used to represent the number of occurrences of a specific LSA_failure_mode that constitutes the threshold or trigger.

Note
An instance of class LSA_failure_mode_related_threshold has no meaning by itself, but only in the context of a specific instance of either:

• Discrete_task_limit, as a trigger
• Discrete_task_limit, as a threshold
• Initial_task_limit, as an initial threshold
• Repeat_task_limit, as a threshold
• Subsequent_repeat_relationship, as a trigger

LSA_failure_mode_related_threshold attributes:

− Event_threshold_number_of_event_occurrences (inherited from class Event_threshold)

LSA_failure_mode_related_threshold associations:

− An instance of LSA_failure_mode_related_threshold is always associated with a specific instance of one of the following:
  • Discrete_task_limit, as a trigger
  • Discrete_task_limit, as a threshold
  • Initial_task_limit, as an initial threshold
  • Repeat_task_limit, as a threshold
  • Subsequent_repeat_relationship, as a trigger

− An association with an instance of LSA_failure_mode

4.15.3.22 Special_event_related_threshold

The Special_event_related_threshold class is a specialization of class Event_threshold. Special_event_related_threshold is used to represent the number of occurrences of a specific Special_event that constitutes the threshold or trigger.

Note
An instance of class Special_event_related_threshold has no meaning by itself, but only in the context of a specific instance of either:

• Discrete_task_limit, as a trigger
• Discrete_task_limit, as a threshold
• Initial_task_limit, as an initial threshold
• Repeat_task_limit, as a threshold
• Subsequent_repeat_relationship, as a trigger

Special_event_related_threshold attributes:

− Event_threshold_number_of_event_occurrences (inherited from class Event_threshold)

Special_event_related_threshold associations:

− An instance of Special_event_related_threshold is always associated with a specific instance of one of the following:
  • Discrete_task_limit, as a trigger
  • Discrete_task_limit, as a threshold
• Initial_task_limit, as an initial threshold
• Repeat_task_limit, as a threshold
• Subsequent_repeat_relationship, as a trigger

− An association with an instance of Special_event

4.15.3.23 Task_related_threshold
The Task_related_threshold class is a specialization of class Event_threshold. Task_related_threshold is used to represent the number of occurrences of a specific Task that constitutes the threshold or trigger.

Note
An instance of class Task_related_threshold has no meaning by itself, but only in the context of a specific instance of either:
• Discrete_task_limit, as a trigger
• Discrete_task_limit, as a threshold
• Initial_task_limit, as an initial threshold
• Repeat_task_limit, as a threshold
• Subsequent_repeat_relationship, as a trigger

Task_related_threshold attributes:
− Event_threshold_number_of_event_occurrences (inherited from class Event_threshold)

Task_related_threshold associations:
− An instance of Task_related_threshold is always associated with a specific instance of one of the following:
  • Discrete_task_limit, as a trigger
  • Discrete_task_limit, as a threshold
  • Periodic_task_limit, as an initial threshold
  • Initial_task_limit, as a threshold
  • Subsequent_repeat_relationship, as a trigger

− An association with an instance of Task

4.15.4 UoF Task Usage (Part 1) - Additions to referenced class and interface definitions
4.15.4.1 Part
The Part class defined in UoF Part, must also implement the following additional interface:
− Task_target_item

4.15.4.2 Breakdown_element_revision
The Breakdown_element_revision class defined in UoF Breakdown Structure, must also implement the following additional interface:
− Task_target_item

4.15.4.3 Rectifying_task
The Rectifying_task class defined in UoF Task, must also implement the following additional interface:
− Planned_task_item

4.15.4.4 Operational_task
The Operational_task class defined in UoF Task, must also implement the following additional interface:
4.15.4.5 Maintenance_level_type
The Maintenance_level_type class defined in UoF Product Usage, must also implement the following additional interface:

- Maintenance_level_allocation_interface

4.15.4.6 Maintenance_location
The Maintenance_location class defined in UoF Product Usage, must also implement the following additional interface:

- Maintenance_level_allocation_interface

4.15.4.7 Operating_location_type
The Operating_location_type class defined in UoF Product Usage, must also implement the following additional interface:

- Maintenance_level_allocation_interface

4.15.4.8 Operating_location
The Operating_location class defined in UoF Product Usage, must also implement the following additional interface:

- Maintenance_level_allocation_interface

4.15.4.9 Special_event
The Special_event class defined in UoF LSA-FMEA and Special Events, must also implement the following additional association:

- An optional association with zero, one or many instances of Special_event_related_threshold

4.15.4.10 LSA_failure_mode
The LSA_failure_mode class defined in UoF LSA-FMEA and Special Events, must also implement the following additional association:

- An optional association with zero, one or many instances of LSA_failure_mode_related_threshold

4.15.4.11 Task
The Task class defined in UoF Task, must also implement the following additional association:

- An optional association with zero, one or many instances of Task_related_threshold

4.16 UoF Task Usage (Part 2)

4.16.1 Overall description
The Task Usage (Part 2) UoF supports the creation of associations between supporting tasks and the objects on which the supporting task is to be performed.
4.16.2 Graphical representation

![Graphical representation of Supporting task usage](ICN-B6865-S3000L0230-001-001)

**Fig 31** UoF Task Usage (Part 2) - class model

4.16.3 UoF Task Usage (Part 2) - New class and interface definitions

4.16.3.1 Supporting_task_usage

The Supporting_task_usage class is a specialization of the Task_usage class, and supports the creation of associations between instances of Supporting_task and specific Task_target_items (ie Breakdown_element_revision or Part) on which the Supporting_task is to be performed.

**Note**
There is one instance of Supporting_task_usage per relevant combination of Supporting_task and Breakdown_element_revision and Part.

**Note**
The class Supporting_task_usage must be used to associate a Supporting_task with the item on which the Supporting_task is to be performed.

Supporting_task_usage associations:

- An association with the specific item (ie instance of Breakdown_element_revision or Part) on which the associated Supporting_task is to be performed (inherited from class Task_usage)
- An optional association with zero, one or many instances of Task_frequency (inherited from class Task_usage)
- An association with the specific Supporting_task which is to be performed on the Task_target_item

4.16.4 UoF Task Usage (Part 2) - Additions to referenced class and interface definitions

4.16.4.1 Supporting_task

The Supporting_task class defined in UoF Task, must also implement the following additional association:

- An association with one or many instances of Supporting_task_usage in order to identify the item(s) on which the Supporting_task is to be performed
4.17 UoF Security Classification
4.17.1 Overall description
The Security Classification UoF supports the assignment of security classes to objects that need special handling with respect to, availability, distribution, presentation etc.

4.17.2 Graphical representation

4.17.3 UoF Security Classification - New class and interface definitions
4.17.3.1 Security_class
The Security_class class identifies security classes that can be assigned to objects within an LSA database.

Security_class attributes:
- Security_class

Security_class associations:
- An optional association with zero, one or many instances of Breakdown_element, Part, Task, Subtask and/or Task_requirement (ie classes that implement the Security_classification_interface)

Note
Security classification system to be used, is determined by the project.

4.17.3.2 Security_classification_interface
The Security_classification_interface is implemented by classes that can be candidates for security classification.

Classes that implement the Security_classification_interface:
- Breakdown_element
− Part
− Task
− Subtask
− Task_requirement

Classes that implement the Security_classification_interface must implement the following association:
− An optional association with zero, one or many instances of Security_class (via the Security_classification class)

4.17.3.3 Security_classification
The Security_classification class defines a relationship between an instance of Security_class and an instance of a class that implements the Security_classification_interface.

Security_classification associations:
− An association with the specific Security_class being applied
− An association with the instance to which the Security_class is applied (an instance of any class that implements the Security_classification_interface)

Note
There is one instance of Security_classification per relevant combination of Security_class and instance to which the Security_class is applied (i.e., relevant instances of Breakdown_element_revision, Part, Task, Subtask or Task_requirement).

Security_classification special recommendations
− Where an instance of Security_classification is dependent upon, e.g., external conditions, customer, etc., it must be distinguished by the assignment of an Applicability_statement (also refer to Para 4.21, UoF Applicability Statement)

4.17.4 UoF Security Classification - Additions to referenced class and interface definitions
4.17.4.1 Breakdown_element
The Breakdown_element class defined in UoF Breakdown Structure, must also implement the following additional interface:
− Security_classification_interface

4.17.4.2 Part
The Part class defined in UoF Part, must also implement the following additional interface:
− Security_classification_interface

4.17.4.3 Task
The Task class defined in UoF Task, must also implement the following additional interface:
− Security_classification_interface

4.17.4.4 Subtask
The Subtask class defined in UoF Task, must also implement the following additional interface:
− Security_classification_interface

4.17.4.5 Task_requirement
The Task_requirement class defined in UoF LSA Candidate Task Requirement, must also implement the following additional interface:
− Security_classification_interface
4.18 UoF Organization Assignment

4.18.1 Overall description
The Organization Assignment UoF supports a flexible way of assigning different types of organizational information to different types of objects in the data model.

Note
There are some previously proposed (even mandatory) associations with the Organization class in the data model. This UoF adds the ability to define any type of additional organizational information which is of relevance for an instance of classes that implement the Organization_assignment_interface. An example of additional organizational information can be 'design_responsible_organization' for a specific part. This is not explicitly defined in the data model but can be added using this UoF, by defining the Organization_assignment_role of class Organization_assignment as 'Design_responsibility'.

4.18.2 Graphical representation

Fig 33 UoF Organization Assignment - class model

4.18.3 UoF Organization Assignment - New class and interface definitions
4.18.3.1 Organization_assignment_interface
The Organization_assignment_interface is implemented by classes that can be associated with optional organizational information.

Classes that implement the Organization_assignment_interface:
− Project
− Contract
− Product
− Product_variant
− Breakdown
− Breakdown_element
− Part
− Material
− Product_variant_realization
− Key_performance_indicator
− Candidate_item_analysis_activity
− Failure_mode
− Change_request
− Task_requirement
− Task
− Resource_specification
− Trade
− Skill
− Task_usage
− Task_limit
− Security_class
− Security_classification
− Document
− Property_representation

Note
Organizations can also be assigned to any class that is a subclass of the classes enumerated above, e.g. to a Rectifying_task which is a subclass of Task (the rule of substitutability).

Classes that implement the Organization_assignment_interface must implement the following association:
− An optional association with zero, one or many instances of Organization (via the Organization_assignment class)

4.18.3.2 Organization_assignment
The Organization_assignment class defines a relationship between a specific Organization and an instance of any class that implements the Organization_assignment_interface.

Note
The role of the Organization being assigned is determined by Organization_assignment_role attribute.

Organization_assignment attributes:
− Organization_assignment_role

Organization_assignment associations:
− An association with the specific Organization being assigned
− An association with the object to which the Organization is assigned (an instance of any class that implements the Organization_assignment_interface)

Note
There is one instance of Organization_assignment per relevant combination of Organization and instance to which the Organization is being assigned (i.e., relevant instance of any class that implements the Organization_assignment_interface).

Organization_assignment special recommendations:
− Where an instance of Organization_assignment is dependent upon eg customer, etc, it must be distinguished by the assignment of an Applicability_statement (also refer to Para 4.21. UoF Applicability Statement)

4.18.4 UoF Organization Assignment - Additions to referenced class and interface definitions

4.18.4.1 Project
The Project class defined in UoF Project, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.2 Contract
The Contract class defined in UoF Project, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.3 Product
The Product class defined in UoF Project, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.4 Product_variant
The Product_variant class defined in UoF Project, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.5 Breakdown
The Breakdown class defined in UoF Breakdown Structure, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.6 Breakdown_element
The Breakdown_element class defined in UoF Breakdown Structure, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.7 Part
The Part class defined in UoF Part, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.8 Material
The Material class defined in UoF Part, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.9 Product_variant_realization
The Product_variant_realization class defined in UoF Product Variant Applicability, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.10 Key_performance_indicator
The Key_performance_indicator class defined in UoF LSA Candidate, must also implement the following additional interface:
4.18.4.11 Candidate_item_analysis_activity
The Candidate_item_analysis_activity class defined in UoF LSA Candidate Analysis Activity, must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.12 Failure_mode
The Failure_mode class defined in UoF LSA-FMEA and Special Events, must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.13 Change_request
The Change_request class defined in UoF LSA Candidate Task Requirement, must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.14 Task_requirement
The Task_requirement class defined in UoF LSA Candidate Task Requirement, must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.15 Task
The Task class defined in UoF Task, must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.16 Resource_specification
The Resource_specification class defined in UoF Task Resources, must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.17 Trade
The Trade class defined in UoF Task Resources, must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.18 Skill
The Skill class defined in UoF Task Resources, must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.19 Task_usage
The Task_usage class defined in UoF Task Usage (Part 1), must also implement the following additional interface:
   - Organization_assignment_interface

4.18.4.20 Task_limit
The Task_limit class defined in UoF Task Usage (Part 1), must also implement the following additional interface:
4.18.4.21 Security_class
The Security_class class defined in UoF Security Classification, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.22 Security_classification
The Security_classification class defined in UoF Security Classification, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.23 Document
The Document class defined in UoF Document, must also implement the following additional interface:
− Organization_assignment_interface

4.18.4.24 Property_representation
The Property_representation class defined in UoF Data Types, must also implement the following additional interface:
− Organization_assignment_interface

Note
The assignment of an Organization to a Property_representation enables the recording of organizational information for any property value in the LSA database, eg in order to identify the Organization that recorded the property value.

4.19 UoF Document
4.19.1 Overall description
The Document UoF supports a flexible way of assigning documents to different types of objects in the data model.

Note
There are some previously proposed (even mandatory) associations with the Document class in the data model. This UoF adds the ability to define any type of additional document information which is of relevance for an instance of classes that implement the Document_assignment_interface. An example of additional document information can be a ‘design document’ for a Part. This is not explicit in the data model but can be added using this UoF, by defining the Document_assignment_role of class Document_assignment as ‘Design_document’.

Note
The S3000L data model defines three specializations of document:

- S1000D data modules and S1000D publication modules, which just contains information that identifies the respective document
- External documents, which contains more meta information about the document itself
4.19.2 Graphical representation

Fig 34 UoF Document - class model
4.19.3 UoF Document - New class and interface definitions
4.19.3.1 Document
The Document class represents documents which are of relevance for the LSA program.

Note
Document is an abstract class, i.e., an instantiation of Document must either be; an S1000D_data_module, an S1000D_publication_module, or an External_document.

Document associations:
- An optional association with zero, one or many instances of any class that implements the Document_assignment_interface (via Document_assignment)

4.19.3.2 S1000D_data_module
The S1000D_data_module class is a specialization of class Document.
The S1000D_data_module class enables the cross references in between the Tasks being defined in S3000L, and its corresponding S1000D data modules, as defined in UoF Task. However, a document that is of type S1000D_data_module can also be assigned to zero, one or many instances of classes that implement the Document_assignment_interface, for the same reason as any other type of document.

S1000D_data_module attributes:
- Data_module_code
- Data_module_issue_number (zero or one)
- Data_module_infoname (zero or one)

S1000D_data_module association:
- An optional association with zero, one or many instances of any class that implements the Document_assignment_interface (inherited from class Document)

4.19.3.3 S1000D_publication_module
The S1000D_publication_module class is a specialization of class Document.
The S1000D_publication_module class enables references from Subtasks to S1000D publication modules, as defined in UoF Task. However, a document that is of type S1000D_publication_module can also be assigned to zero, one or many instances of classes that implement the Document_assignment_interface, for the same reason as any other type of document.

S1000D_publication_module attributes:
- Publication_module_code
- Publication_module_issue_number (zero or one)
- Publication_module_title (zero or one)

S1000D_publication_module association:
- An optional association with zero, one or many instances of any class that implements the Document_assignment_interface (inherited from class Document)

4.19.3.4 External_document
The External_document class is a specialization of class Document, and represents all documents that are not identified and produced in accordance with S1000D.

External_document attributes:
- Document_identifier (one or many)
- Document_title
- Document_issue_identifier (zero or one)
- Document_issue_date (zero or one)
- Document_type (zero or one)
- Document_location (zero or one)

External_document associations:

- An optional association with zero, one or many instances of any class that implements the Document_assignment_interface (inherited from class Document)

4.19.3.5 Document_assignment_interface

The Document_assignment_interface is implemented by classes that can be associated with optional document information.

Classes that implement the Document_assignment_interface:

- Project
- Contracted_product_variant
- Contract
- Product
- Product_variant
- Organization
- Operating_location
- Maintenance_location
- Operating_location_type
- Maintenance_level_type
- Breakdown
- Breakdown_revision
- Breakdown_element
- Breakdown_element_revision
- Part
- Material
- Hardware_part_operational_authorised_life
- Alternate_part_relationship
- Parts_list_entry
- Hardware_element_realization
- Software_element_realization
- Hardware_element_in_zone_relationship
- Product_variant_realization
- Key_performance_indicator
- Mean_time_between_failure_correction
- Failure_rate_correction
- Candidate_item_analysis_activity
- LSA_candidate_technology_behavior_rating
- LSA_failure_mode
- Failure_mode
- Failure_mode_effect
- Special_event
- Product_usage_phase
- Detectability
- Task_requirement
- Change_request
- Task
- Warning_caution_or_note
- Task_change
- Subtask
- Subtask_timeline
- Subtask_acceptance_parameter
- Circuit_breaker
- Task_resource
- Skill
- Trade
- Special_training_requirement
- Resource_specification
- Resource_realization
- Task_usage
- Task_frequency
- Task_limit
- Security_class
- Security_classification
- Organization_assignment
- Remark
- Applicability_statement
- Condition_by_identifier
- Condition_type
- Class
- Prp

**Note**
Documents can also be assigned to any class that is a subclass of the classes enumerated above, e.g., to a Rectifying_task which is a subclass of Task (the rule of substitutability).

Classes that implement the Document_assignment_interface must implement the following association:

- An optional association with zero, one or many instances of Documents (via the Document_assignment class)

### 4.19.3.6 Document_assignment

The Document_assignment class defines a relationship between a specific Document and an instance of any class that implements the Document_assignment_interface.

**Note**
The role of the document being assigned is determined by Document_assignment_role attribute.

**Note**
The Document_portion attribute identifies a defined portion of a document which is of interest in a specific usage.

**Document_assignment attributes:**

- Document_assignment_role
- Document_portion (zero or one)

**Document associations:**

- An association with the specific Document being assigned
- An association with the object to which the instance of Document is being assigned (i.e., an instance of any class that implements the Document_assignment_interface)
Note

There is one instance of Document_assignment per relevant combination of Document instance and instance to which the Document is assigned (i.e., instance of any class that implements the Document_assignment_interface).

Document_assignment special recommendations:

− Where an instance of Document_assignment is dependent upon, e.g., customer, etc., it must be distinguished by the assignment of an Applicability_statement (also refer to Para 4.21, UoF Applicability Statement)

4.19.4 UoF Document - Additions to referenced class and interface definitions

4.19.4.1 Project
The Project class defined in UoF Project must also implement the following additional interface:

− Document_assignment_interface

4.19.4.2 Contracted_product_variant
The Contracted_product_variant class defined in UoF Project must also implement the following additional interface:

− Document_assignment_interface

4.19.4.3 Contract
The Contract class defined in UoF Project must also implement the following additional interface:

− Document_assignment_interface

4.19.4.4 Product
The Product class defined in UoF Project must also implement the following additional interface:

− Document_assignment_interface

4.19.4.5 Product_variant
The Product_variant class defined in UoF Project must also implement the following additional interface:

− Document_assignment_interface

4.19.4.6 Organization
The Organization class defined in UoF Project must also implement the following additional interface:

− Document_assignment_interface

4.19.4.7 Operating_location
The Operating_location class defined in UoF Product Usage must also implement the following additional interface:

− Document_assignment_interface

4.19.4.8 Maintenance_location
The Maintenance_location class defined in UoF Product Usage must also implement the following additional interface:

− Document_assignment_interface
4.19.4.9 Maintenance_level_type
The Maintenance_level_type class defined in UoF Product Usage must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.10 Operating_location_type
The Operating_location_type class defined in UoF Product Usage must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.11 Breakdown
The Breakdown class defined in UoF Breakdown Structure must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.12 Breakdown_revision
The Breakdown_revision class defined in UoF Breakdown Structure must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.13 Breakdown_element
The Breakdown_element class defined in UoF Breakdown Structure must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.14 Breakdown_element_revision
The Breakdown_element_revision class defined in UoF Breakdown Structure must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.15 Part
The Part class defined in UoF Part must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.16 Material
The Material class defined in UoF Part must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.17 Hardware_part_operational_authorized_life
The Hardware_part_operational_authorized_life class defined in UoF Part must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.18 Alternate_part_relationship
The Alternate_part_relationship class defined in UoF Part must also implement the following additional interface:
   - Document_assignment_interface
4.19.4.19 Parts_list_entry
The Parts_list_entry class defined in UoF Part must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.20 Hardware_element_realization
The Hardware_element_realization class defined in UoF Breakdown Element Realization must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.21 Software_element_realization
The Software_element_realization class defined in UoF Breakdown Element Realization must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.22 Hardware_element_in_zone_relationship
The Hardware_element_in_zone_relationship class defined in UoF Breakdown Zone Element must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.23 Product_variant_realization
The Product_variant_realization class defined in UoF Product Variant Applicability must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.24 Key_performance_indicator
The Key_performance_indicator class defined in UoF LSA Candidate must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.25 Mean_time_between_failure_correction
The Mean_time_between_failure_correction class defined in UoF LSA Candidate must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.26 Failure_rate_correction
The Failure_rate_correction class defined in UoF LSA Candidate must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.27 Candidate_item_analysis_activity
The Candidate_item_analysis_activity class defined in UoF LSA Candidate Analysis Activity must also implement the following additional interface:
   - Document_assignment_interface

4.19.4.28 LSA_candidate_technology_behavior_rating
The LSA_candidate_technology_behavior_rating class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
   - Document_assignment_interface
4.19.4.29 LSA_failure_mode
The LSA_failure_mode class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
− Document_assignment_interface

4.19.4.30 Failure_mode
The Failure_mode class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
− Document_assignment_interface

4.19.4.31 Failure_mode_effect
The Failure_mode_effect class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
− Document_assignment_interface

4.19.4.32 Special_event
The Special_event class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
− Document_assignment_interface

4.19.4.33 Product_usage_phase
The Product_usage_phase class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
− Document_assignment_interface

4.19.4.34 Detectability
The Detectability class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
− Document_assignment_interface

4.19.4.35 Task_requirement
The Task_requirement class defined in UoF LSA Candidate Task Requirement must also implement the following additional interface:
− Document_assignment_interface

4.19.4.36 Change_request
The Change_request class defined in UoF LSA Candidate Task Requirement must also implement the following additional interface:
− Document_assignment_interface

4.19.4.37 Task
The Task class defined in UoF Task must also implement the following additional interface:
− Document_assignment_interface

4.19.4.38 Warning_caution_or_note
The Warning_caution_or_note class defined in UoF Task must also implement the following additional interface:
− Document_assignment_interface
4.19.4.39 Task_change
The Task_change class defined in UoF Task must also implement the following additional interface:
- Document_assignment_interface

4.19.4.40 Subtask
The Subtask class defined in UoF Task must also implement the following additional interface:
- Document_assignment_interface

4.19.4.41 Subtask_timeline
The Subtask_timeline class defined in UoF Task must also implement the following additional interface:
- Document_assignment_interface

4.19.4.42 Subtask_acceptance_parameter
The Subtask_acceptance_parameter class defined in UoF Task must also implement the following additional interface:
- Document_assignment_interface

4.19.4.43 Circuit_breaker
The Circuit_breaker class defined in UoF Task must also implement the following additional interface:
- Document_assignment_interface

4.19.4.44 Task_resource
The Task_resource class defined in UoF Task Resources must also implement the following additional interface:
- Document_assignment_interface

4.19.4.45 Skill
The Skill class defined in UoF Task Resources must also implement the following additional interface:
- Document_assignment_interface

4.19.4.46 Trade
The Trade class defined in UoF Task Resources must also implement the following additional interface:
- Document_assignment_interface

4.19.4.47 Special_training_requirement
The Special_training_requirement class defined in UoF Task Resources must also implement the following additional interface:
- Document_assignment_interface

4.19.4.48 Resource_specification
The Resource_specification class defined in UoF Task Resources must also implement the following additional interface:
- Document_assignment_interface
4.19.4.49 Resource_realization
The Resource_realization class defined in UoF Task Resources must also implement the following additional interface:
- Document_assignment_interface

4.19.4.50 Task_usage
The Task_usage class defined in UoF Task Usage (Part 1) must also implement the following additional interface:
- Document_assignment_interface

4.19.4.51 Task_frequency
The Task_frequency class defined in UoF Task Usage (Part 1) must also implement the following additional interface:
- Document_assignment_interface

4.19.4.52 Task_limit
The Task_limit class defined in UoF Task Usage (Part 1) must also implement the following additional interface:
- Document_assignment_interface

4.19.4.53 Security_class
The Security_class class defined in UoF Security Classification must also implement the following additional interface:
- Document_assignment_interface

4.19.4.54 Security_classification
The Security_classification class defined in UoF Security Classification must also implement the following additional interface:
- Document_assignment_interface

4.19.4.55 Organization_assignment
The Organization_assignment class defined in UoF Organization Assignment must also implement the following additional interface:
- Document_assignment_interface

4.19.4.56 Remark
The Remark class defined in UoF Remark must also implement the following additional interface:
- Document_assignment_interface

4.19.4.57 Applicability_statement
The Applicability_statement class defined in UoF Applicability Statement must also implement the following additional interface:
- Document_assignment_interface

4.19.4.58 Condition_by_identifier
The Condition_by_identifier class defined in UoF Applicability Statement must also implement the following additional interface:
- Document_assignment_interface
4.19.4.59 Condition_type
The Condition_type class defined in UoF Applicability Statement must also implement the following additional interface:

- Document_assignment_interface

4.19.4.60 Class
The Class class defined in UoF Data Types must also implement the following additional interface:

- Document_assignment_interface

4.19.4.61 Prp
The Prp class defined in UoF Data Types must also implement the following additional interface:

- Document_assignment_interface

4.20 UoF Remark
4.20.1 Overall description
The Remark UoF supports a flexible way of assigning remarks to instances of almost any class in the data model.
4.20.2 Graphical representation

4.20.3 UoF Remark - New class and interface definitions

Remark

The Remark class represents remarks of any kind, which anyone working within the LSA Program would like to put against an object of interest.

Remark attributes:

Applicable to: All
Remark_text
Remark_type (zero or one)

Note
Since the Remark_text is of data type Descr, all remarks also identify the organization that provided the remark, and the date when it was recorded.

Remark associations:
- An association with one or many instances of any class that implements the Remark_assignment_interface

4.20.3.2 Remark_assignment_interface
The Remark_assignment_interface is implemented by classes that can be associated with optional remarks.

Classes that must implement the Remark_assignment_interface:
- Project
- Contract
- Product
- Product_variant
- Contracted_product_variant
- Organization
- Contracted_product_variant_in_operating_location
- Contracted_product_variant_in_operating_location_type
- Maintenance_level_type
- Operating_location
- Operating_location_type
- Maintenance_location
- Breakdown
- Breakdown_revision
- Breakdown_element
- Breakdown_element_revision
- Breakdown_element_relationship
- Part
- Parts_list_entry
- Hardware_part_operationalAuthorized_life
- Material
- Hardware_part_material_usage
- Alternate_part_relationship
- Substitute_part_relationship
- Hardware_element_realization
- Software_element_realization
- Hardware_element_in_zone_relationship
- Product_variant_realization
- Product_variant_applicability
- Product_variant_realization_applicability
- Key_performance_indicator
- Failure_rate_correction
- Mean_time_between_failure_correction
- Candidate_item_analysis_activity
- Failure_mode
- LSA_failure_mode
- Special_event
- Failure_mode_effect
- LSA_candidate_technology_behavior_rating
- Special_event_occurrence_definition
- Product_usage_phase
- Special_event_effects
- Detectability
- Task_requirement
- Task_requirement_change
- Change_request
- Task
- Task_distribution
- Warning_caution_or_note
- Task_change
- Subtask
- Subtask_timeline
- Subtask_in_zone
- Subtask_acceptance_parameter
- Subtask_objective_state
- Circuit_breaker
- Task_resource
- Task_resource_relationship
- Resource_specification
- Resource_realization
- Special_training_requirement
- Skill
- Trade
- Task_usage
- Task_frequency
- Maintenance_level_allocation
- Task_limit
- Sampling
- Security_class
- Security_classification
- Document
- Document_assignment
- Organization_assignment
- Applicability_statement
- Condition_by_identifier
- Condition_type
- Condition_statement
- Serial_number_range
- Class
- Classification
- Prp
- Property_representation

Note
Remarks can also be assigned to any class that is a subclass of the classes enumerated above, eg to a Rectifying_task which is a subclass of Task (the rule of substitutability).

Classes that implement the Remark_assignment_interface must implement the following association:
- An optional association with zero, one or many remarks
4.20.4 UoF Remark - Additions to referenced class and interface definitions

4.20.4.1 Project
The Project class defined in UoF Project must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.2 Contract
The Contract class defined in UoF Project must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.3 Product
The Product class defined in UoF Project must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.4 Product_variant
The Product_variant class defined in UoF Project must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.5 Contracted_product_variant
The Contracted_product_variant class defined in UoF Project must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.6 Organization
The Organization class defined in UoF Project must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.7 Contracted_product_variant_in_operating_location
The Contracted_product_variant_in_operating_location class defined in UoF Product Usage must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.8 Contracted_product_variant_in_operating_location_type
The Contracted_product_variant_in_operating_location_type class defined in UoF Product Usage must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.9 Maintenance_level_type
The Maintenance_level_type class defined in UoF Product Usage must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.10 Operating_location
The Operating_location class defined in UoF Product Usage must also implement the following additional interface:
- Remark_assignment_interface
4.20.4.11 Operating_location_type
The Operating_location_type class defined in UoF Product Usage must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.12 Maintenance_location
The Maintenance_location class defined in UoF Product Usage must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.13 Breakdown
The Breakdown class defined in UoF Breakdown Structure must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.14 Breakdown_revision
The Breakdown_revision class defined in UoF Breakdown Structure must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.15 Breakdown_element
The Breakdown_element class defined in UoF Breakdown Structure must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.16 Breakdown_element_revision
The Breakdown_element_revision class defined in UoF Breakdown Structure must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.17 Breakdown_element_relationship
The Breakdown_element_relationship class defined in UoF Breakdown Structure must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.18 Part
The Part class defined in UoF Part must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.19 Parts_list_entry
The Parts_list_entry class defined in UoF Part must also implement the following additional interface:
  - Remark_assignment_interface

4.20.4.20 Hardware_part_operational_authorised_life
The Hardware_part_operational_authorised_life class defined in UoF Part must also implement the following additional interface:
  - Remark_assignment_interface
4.20.4.21 Material
The Material class defined in UoF Part must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.22 Hardware_part_material_usage
The Hardware_part_material_usage class defined in UoF Part must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.23 Alternate_part_relationship
The Alternate_part_relationship class defined in UoF Part must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.24 Substitute_part_relationship
The Substitute_part_relationship class defined in UoF Part must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.25 Hardware_element_realization
The Hardware_element_realization class defined in UoF Breakdown Element Realization must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.26 Software_element_realization
The Software_element_realization class defined in UoF Breakdown Element Realization must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.27 Hardware_element_in_zone_relationship
The Hardware_element_in_zone_relationship class defined in UoF Breakdown Zone Element must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.28 Product_variant_realization
The Product_variant_realization class defined in UoF Product Variant Applicability must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.29 Product_variant_applicability
The Product_variant_applicability class defined in UoF Product Variant Applicability must also implement the following additional interface:
   - Remark_assignment_interface

4.20.4.30 Product_variant_realization_applicability
The Product_variant_realization_applicability class defined in UoF Product Variant Applicability must also implement the following additional interface:
   - Remark_assignment_interface
4.20.4.31 Key_performance_indicator
The Key_performance_indicator class defined in UoF LSA Candidate must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.32 Failure_rate_correction
The Failure_rate_correction class defined in UoF LSA Candidate must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.33 Mean_time_between_failure_correction
The Mean_time_between_failure_correction class defined in UoF LSA Candidate must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.34 Candidate_item_analysis_activity
The Candidate_item_analysis_activity class defined in UoF LSA Candidate Analysis Activity must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.35 Failure_mode
The Failure_mode class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.36 LSA_failure_mode
The LSA_failure_mode class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.37 Special_event
The Special_event class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.38 Failure_mode_effect
The Failure_mode_effect class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.39 LSA_candidate_technology_behavior_rating
The LSA_candidate_technology_behavior_rating class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
   – Remark_assignment_interface

4.20.4.40 Special_event_occurrence_definition
The Special_event_occurrence_definition class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
   – Remark_assignment_interface
4.20.4.41 Product_usage_phase
The Product_usage_phase class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.42 Special_event_effects
The Special_event_effects class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.43 Detectability
The Detectability class defined in UoF LSA-FMEA and Special Events must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.44 Task_requirement
The Task_requirement class defined in UoF LSA Candidate Task Requirement must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.45 Task_requirement_change
The Task_requirement_change class defined in UoF LSA Candidate Task Requirement must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.46 Change_request
The Change_request class defined in UoF LSA Candidate Task Requirement must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.47 Task
The Task class defined in UoF Task must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.48 Task_distribution
The Task_distribution class defined in UoF Task must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.49 Warning_caution_or_note
The Warning_caution_or_note class defined in UoF Task must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.50 Task_change
The Task_change class defined in UoF Task must also implement the following additional interface:
- Remark_assignment_interface
4.20.4.51 Subtask
The Subtask class defined in UoF Task must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.52 Subtask_timeline
The Subtask_timeline class defined in UoF Task must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.53 Subtask_in_zone
The Subtask_in_zone class defined in UoF Task must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.54 Subtask_acceptance_parameter
The Subtask_acceptance_parameter class defined in UoF Task must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.55 Subtask_objective_state
The Subtask_objective_state class defined in UoF Task must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.56 Circuit_breaker
The Circuit_breaker class defined in UoF Task must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.57 Task_resource
The Task_resource class defined in UoF Task Resources must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.58 Task_resource_relationship
The Task_resource_relationship class defined in UoF Task Resources must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.59 Resource_specification
The Resource_specification class defined in UoF Task Resources must also implement the following additional interface:
− Remark_assignment_interface

4.20.4.60 Resource_realization
The Resource_realization class defined in UoF Task Resources must also implement the following additional interface:
− Remark_assignment_interface
4.20.4.61 Special_training_requirement
The Special_training_requirement class defined in UoF Task Resources must also implement
the following additional interface:

− Remark_assignment_interface

4.20.4.62 Skill
The Skill class defined in UoF Task Resources must also implement the following additional
interface:

− Remark_assignment_interface

4.20.4.63 Trade
The Trade class defined in UoF Task Resources must also implement the following additional
interface:

− Remark_assignment_interface

4.20.4.64 Task_usage
The Task_usage class defined in UoF Task Usage (Part 1) must also implement the following additional
interface:

− Remark_assignment_interface

4.20.4.65 Task_frequency
The Task_frequency class defined in UoF Task Usage (Part 1) must also implement the following additional
interface:

− Remark_assignment_interface

4.20.4.66 Maintenance_level_allocation
The Maintenance_level_allocation class defined in UoF Task Usage (Part 1) must also implement the following additional interface:

− Remark_assignment_interface

4.20.4.67 Task_limit
The Task_limit class defined in UoF Task Usage (Part 1) must also implement the following additional interface:

− Remark_assignment_interface

4.20.4.68 Sampling
The Sampling class defined in UoF Task Usage (Part 1) must also implement the following additional interface:

− Remark_assignment_interface

4.20.4.69 Security_class
The Security_class class defined in UoF Security Classification must also implement the following additional interface:

− Remark_assignment_interface

4.20.4.70 Security_classification
The Security_classification class defined in UoF Security Classification must also implement the following additional interface:

− Remark_assignment_interface
4.20.4.71 Document
The Document class defined in UoF Document must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.72 Document_assignment
The Document_assignment class defined in UoF Document must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.73 Organization_assignment
The Organization_assignment class defined in UoF Organization Assignment must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.74 Applicability_statement
The Applicability_statement class defined in UoF Applicability Statement must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.75 Condition_by_identifier
The Condition_by_identifier class defined in UoF Applicability Statement must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.76 Condition_type
The Condition_type class defined in UoF Applicability Statement must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.77 Condition_statement
The Condition_statement class defined in UoF Applicability Statement must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.78 Serial_number_range
The Serial_number_range class defined in UoF Applicability Statement must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.79 Class
The Class class defined in UoF Data Types must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.80 Classification
The Classification class defined in UoF Data Types must also implement the following additional interface:
- Remark_assignment_interface
4.20.4.81 Prp
The Prp class defined in UoF Data Types must also implement the following additional interface:
- Remark_assignment_interface

4.20.4.82 Property_representation
The Property_representation class defined in UoF Data Types must also implement the following additional interface:
- Remark_assignment_interface

4.21 UoF Applicability Statement
4.21.1 Overall description
The Applicability Statement UoF defines a flexible way of assigning applicability statements to instances of relevant classes in the data model. This UoF supports simple to highly complex definitions of applicability statements.

Note
The S3000L applicability statement class model is based on the applicability model defined in S1000D.

The Applicability Statement class model supports:
- Identification of parameters that can affect the applicability of LSA data (eg parameters such as "model A", "tail number 15", "arctic", "desert", "land-based", "sea-based", etc)
- Creation of computable logical expressions which can be evaluated, using the identified parameters
- Enumeration of the S3000L classes that can have an associated applicability statement

The definition of an applicability statement is based upon the following constructs:
- Applicability assignment, which identifies classes to which an applicability statement can be assigned
- Applicability evaluation, which defines the expression to be evaluated in order for the target item to be applicable

The applicability expression to be evaluated can be either a logical expression, or a value to be asserted. Any use of a logical expression will in turn relate to one or many values which are to be asserted.

Note
S1000D defines two separate types of parameters which can affect the applicability of tech data. Firstly, product attributes and secondly, conditions. Product attributes are properties of the product that are typically set at the time of manufacture of a product instance and will not change throughout the service life of a product instance. Examples of product attributes given in S1000D are 'serial number' or 'model'. Conditions may be technical, operational, environmental or any other type that can change throughout the service life of a product instance. Examples of conditions given in S1000D are location of maintenance, temperature, wind speed and sandy conditions. However, there is no hard and fast rule in S1000D for distinguishing between product attributes and conditions, which is why S3000L does not implement this distinction in the data model. Mapping in between S3000L parameters and S1000D product attributes and conditions needs to be done on a per project basis.

The Applicability Cross Reference (ACT) and Condition Cross Reference (CCT) data modules in S1000D can be generated from the defined Conditions in S3000L, and/or from the instances that defined for classes that implements the Applicability_assert_interface.
Applicability statements in S1000D (<applic>) can be generated from Applicability statements defined in S3000L. Basic rules are:

- The S1000D <applic> element can be derived from the Applicability statement section of the S3000L data model, which includes Applicability_statement, Dated_applicability_statement and the Applicability_assignment interface.
- The content of the S1000D <evaluate> element within the <applic> element can be derived from the Applicability evaluation section of the S3000L data model, which includes:
  - Applicability_evaluation_by_logical_operator
  - Logical_operator specializations And, Or, XOR and NOT
- The content of the S1000D <assert> element within the <applic>/<evaluate> elements can be derived from:
  - Applicability_evaluation_by_assertion_of_class_instance together with instances of classes that implements the Applicability_assert_interface
  - Applicability_evaluation_by_assertion_of_condition together with Condition_statement which relates to a defined Condition_definition and a Condition_type_value
  - Applicability_evaluation_by_applicability_statement_reference which relates to a previously defined Applicability_statement.

Note
Applicability_statement is used to define usage related restrictions, whereas Product_variant_applicability and Product_variant_realization_applicability (UoF Product Variant Applicability) are used to define restrictions for a given product design (product definition).
4.21.2 Graphical representation

Fig 36 UoF Applicability Statement - class model
4.21.3 UoF Applicability Statement - New class and interface definitions

4.21.3.1 Condition_type

The Condition_type class defines types of conditions to be used in applicability statements, where the specific type of condition cannot be derived from any other class that is defined within the S3000L data model.

Note
Examples of condition types that cannot be derived from the use of any other class in the S3000L data model, and therefore needs to be defined as instances of Condition_type, are:

- Service bulletin
- Operational environment
- Wind speed
- Ashore/Atfloat
- Operational scenario
- Operational state

Note
An example of a class in the data model which can be used to define an Applicability_evaluation_by_assertion expression is the Customer class. The values that can be used in the Applicability_evaluation_by_assertion_of_class_instance are the identifier values for the instances of the Customer class, ie the values already exist in the LSA database.

Condition_type attributes:

- Condition_type_name
- Condition_type_definition (zero or one)

The Condition_type class must implement the following interface:

- Condition_definition

Condition_type associations:

- An association with one or many Condition_type_values
- An optional association with zero, one or many instances of Condition_by_identifier, where the Condition_by_identifier is an instantiation of the Condition_type

Note
An example on Condition_type and Condition_by_identifier is ‘Service Bulletin’ (as an instance of Condition_type) and ‘Service bulletin S001 - Chain guard’ (as an instance of Condition_by_identifier)

Note
The following rules can be applied to populate an S1000D Condition Cross-reference Table (CCT):

- The value of Condition_type_name can populate the <conditiontype id> attribute and <conditiontype><name> element, respectively
- The value of Condition_type_definition can be left empty if the definition of the term stated as the Condition_type_name is defined in a Reference Data Library (RDL). Depending on where the definition of the Condition_type resides, the S1000D <conditiontype><description> element can be populated either by the value given in Condition_type_definition or the value available in the referenced RDL

4.21.3.2 Condition_type_value

The Condition_type_value class defines valid values for a specific Condition_type.
Note
The Condition_type_value class is an abstract class, i.e. any of its subclasses must be used for the actual representation of the expression to be evaluated or asserted.

- Condition_type_value subclasses:
  - Condition_type_class_value
  - Condition_type_property_value

Note
Instances of the Condition_type_value have no meaning by itself, but only in the context of a specific Condition_type, or a specific Condition_statement.

Condition_type_value associations:
- An instance of Condition_type_value is always associated with either an instance of Condition_type, or an instance of Condition_statement

4.21.3.3 Condition_type_class_value
Instances of Condition_type_class_value must refer to valid Reference Data Library classes.

Note
Examples on valid class names that can be used are:
- ‘Pre’/’Post’ defined as valid values for e.g. the ‘Service_bulletin’ Condition_type
- ‘Desert’/’Arctic’ defined as valid values for ‘Operational_environment’ Condition_type

Note
Instances of the Condition_type_class_value have no meaning by itself, but only in the context of a specific Condition_type, or a specific Condition_statement.

Condition_type_class_value attributes:
- Condition_type_class_value

Condition_type_class_value associations:
- An instance of Condition_type_class_value is always associated with either an instance of Condition_type, or an instance of Condition_statement (inherited from class Condition_type_value)

Note
The following rules can be applied to populate an S1000D Condition Cross-reference Table (CCT):
- The value of Condition_type_class_value can populate the <conditiontype><enum> element for the associated Condition_type

4.21.3.4 Condition_type_property_value
The Condition_type_property_value class defines valid values, value ranges etc, which can be used for a specific Condition_type.

Note
Valid values can be of any kind of Property_representation, i.e. it can be a Value_with_unit_property, Value_with_tolerances_property, Value_range_property or a Value_with_limit_property.

Note
Examples on valid property values are ‘0-15’, ‘15-30’ and ‘30-45’ defined for Condition_type ‘Wind_speed’.
Note

Instances of the Condition_type_property_value have no meaning by itself, but only in the context of a specific Condition_type, or a specific Condition_statement.

Condition_type_property_value attributes:
− Condition_type_property_value

Condition_type_property_value associations:
− An instance of Condition_type_property_value is always associated with either an instance of Condition_type, or an instance of Condition_statement (inherited from class Condition_type_value)

Note

The following rules can be applied to populate an S1000D Condition Cross-reference Table (CCT):

- The value of Condition_type_property_value can populate the <conditiontype><enum> element for the associated Condition_type

4.21.3.5 Condition_definition

The Condition_definition interface is implemented by the condition classes that can be asserted in order to determine the applicability of the data being defined during LSA.

Classes that implement the Condition_definition interface:
− Condition_type
− Condition_by_identifier

Classes that implement the Condition_definition interface must implement the following associations:
− An optional association with one or many instances of Condition_statement

4.21.3.6 Condition_by_identifier

The Condition_by_identifier class identifies specific instances of a condition type that can be used as conditions in one or many applicability statements.

Note:

An example of an identified condition is:

- Service bulletin S001 - Chain guard (an identified instance of the service bulletin Condition_type)

Condition_by_identifier attributes:
− Condition_identifier
− Condition_name
− Condition_description (zero or one)

The Condition_by_identifier class must implement the following interface:
− Condition_definition

Condition_by_identifier associations:
− An instance of Condition_by_identifier is always associated with a specific Condition_type

Note

The following rules can be applied to populate an S1000D Condition Cross-reference Table (CCT):
• The value of Condition_identifier can populate the <condition id> attribute
• The value of Condition_description can populate the <condition><description> element
• The value of Condition_name can populate the <condition><name> and <condition><displayname> elements, respectively

4.21.3.7 Applicability_statement

The Applicability_statement class defines explicit applicability statements to be applied against instances in the LSA database.

Note
An Applicability_statement can be evaluated in a maintenance execution environment, eg using an Interactive Electronic Technical Publishing (IETP).

Applicability_statement attributes:

− Applicability_statement_identifier (zero or one)
− Applicability_statement_description (zero or one)

Applicability_statement associations:

− An optional association with zero or one computer interpretable expression, ie an instance of Applicability_evaluation
− An association with one or many objects for which the Applicability_statement is valid, ie instances of any class that implements the Applicability_assignment interface
− An optional association with zero, one or many instances of Applicability_evaluation_by_applicability_statement_reference, in which the Applicability_statement can be asserted as part of another Applicability_evaluation.

Note
The Applicability_statement class can be specialized into a Dated_applicability_statement.

4.21.3.8 Dated_applicability_statement

The Dated_applicability_statement class is a specialization of Applicability_statement, and defines explicit applicability statements to be used against instances in the LSA database, where the applicability is not just defined by the expression to be evaluated, but is also limited in time.

Dated_applicability_statement attributes:

− Applicability_statement_identifier (inherited from class Applicability_statement)
− Applicability_statement_description (inherited from class Applicability_statement)
− Applicability_start_date
− Applicability_end_date (zero or one)

Dated_applicability_statement associations:

− An optional association with zero or one computer interpretable expression, ie an instance of Applicability_evaluation (inherited from class Applicability_statement)
− An association with one or many objects for which the Applicability_statement is valid, ie instances of any class that implements the Applicability_assignment interface (inherited from class Applicability_statement)
− An optional association with zero, one or many instances of Applicability_evaluation_by_applicability_statement_reference, in which the Applicability_statement can be asserted as part of another Applicability_evaluation (inherited from class Applicability_statement)

4.21.3.9 Applicability_assignment

The Applicability_assignment interface is implemented by classes that can be associated with optional applicability statements.
Classes that must implement the Applicability_assignment interface:

- Substitute_part_relationship
- Alternate_part_relationship
- Hardware_part_operational_authorized_life
- Parts_list_entry
- Key_performance_indicator
- Failure_rate_correction
- Mean_time_between_failure_correction
- LSA_failure_mode
- Failure_mode
- Special_event_occurrence_definition
- Failure_mode_effect
- Special_event_effects
- Task_requirement
- Task
- Subtask
- Subtask_timeline
- Subtask_in_zone
- Task_warning_caution_or_note
- Subtask_warning_caution_or_note
- Subtask_acceptance_parameter
- Subtask_objective_state
- Task_resource
- Task_resource_relationship
- Resource_realization
- Task_personnel_resource_competence
- Task_usage
- Task_frequency
- Maintenance_level_allocation
- Task_limit
- Threshold_definition
- Security_classification
- Document_assignment
- Organization_assignment
- Classification
- Class
- Descr
- Prp

**Note**

Applicability_assignment can also be assigned to any class that is a subclass of the classes enumerated above, eg Rectifying_task which is a subclass of Task (rule of substitutability).

Classes that implement the Applicability_assignment interface must implement the following association:

- An optional association with one instance of Applicability_statement per instance of the respective class

### 4.21.3.10 Applicability_evaluation

The Applicability_evaluation class identifies an expression against which an applicability statement can be evaluated or asserted.

The expression to be evaluated is either a logical expression, or a value to be asserted. These are defined as separate subclasses of Applicability_evaluation.
Note
The Applicability_evaluation class is an abstract class, i.e., any of its subclasses must be used for the actual representation of the expression to be evaluated or asserted.

Note
Instances of Applicability_evaluation don’t have any meaning outside the context of the Applicability_statement or the Logical_operator in which it is being used.

Applicability_evaluation associations:
- An instance of Applicability_evaluation is always associated with an instance of Applicability_statement, either directly or as a logical operator operand (i.e., as part of an AND, OR, XOR or NOT expression).

4.21.3.11 Applicability_evaluation_by_logical_operator
The Applicability_evaluation_by_logical_operator class is a specialization of Applicability_evaluation, and defines a logical expression against which an applicability statement can be evaluated.

Note
All Logical_operators that are part of a Applicability_evaluation_by_logical_operator refers to other instances of Applicability_evaluation, which in turn can be either a logical expression or a value to be asserted, and so on.

Note
Instances of Applicability_evaluation_by_logical_operator don’t have any meaning outside the context of the Applicability_statement or Logical_operator, in which it is being defined.

Applicability_evaluation_by_logical_operator associations:
- An instance of Applicability_evaluation_by_logical_operator is always associated with a specific Applicability_statement, either directly, or as a logical operator operand (i.e., as part of an AND, OR, XOR or NOT expression).
- An instance of Applicability_evaluation_by_logical_operator must be associated with an instance of a Logical_operator which in turn refers to other instances of Applicability_evaluation to be evaluated or asserted.

4.21.3.12 Logical_operator
The Logical_operator class defines the logical expression to be evaluated.

Note
Logical_operator is an abstract class, i.e., any instantiation of Logical_operator must be an instance of either AND, OR, XOR or NOT.

Logical_operator associations:
- An instance of Logical_operator is always associated with a specific instance of Applicability_evaluation_by_logical_operator.

4.21.3.13 AND
The AND class is a specialization of Logical_operator, and defines a list of Applicability_evaluations that all need to be TRUE in order for the overall Applicability_statement to be TRUE.

Note
S1000D uses AND statements in the definition of an <applic> element.

AND associations:
An instance of AND is always associated with a specific instance of Applicability_evaluation_by_logical_operator (inherited from class Logical_operator)
An instance of AND relates to two or more instances of Applicability_evaluation as its operands

4.21.3.14 OR
The OR class is a specialization of Logical_operator, and defines a list of Applicability_evaluations where at least one needs to be TRUE in order for the overall Applicability_statement to be TRUE.

Note
S1000D does not use OR statements in the definition of an <applic> element.

OR associations:
An instance of OR is always associated with a specific instance of Applicability_evaluation_by_logical_operator (inherited from class Logical_operator)
An instance of OR relates to two or more instances of Applicability_evaluation as its operands

4.21.3.15 XOR
The XOR class is a specialization of Logical_operator, and defines a list of Applicability_evaluations where exactly one can be TRUE in order for the overall Applicability_statement to be TRUE.

Note
S1000D does not use XOR statements in the definition of an <applic> element.

XOR associations:
An instance of XOR is always associated with a specific instance of Applicability_evaluation_by_logical_operator (inherited from class Logical_operator)
An instance of XOR relates to two or more instances of Applicability_evaluation as its operands

4.21.3.16 NOT
The NOT class is a specialization of Logical_operator, and defines an Applicability_evaluations that needs to be FALSE in order for the overall Applicability_statement to be TRUE.

Note
S1000D does not make explicit use of NOT statements in the definition of an <applic> element.

NOT associations:
An instance of NOT is always associated with a specific instance of Applicability_evaluation_by_logical_operator (inherited from class Logical_operator)
An instance of NOT relates to one instance of Applicability_evaluation as its operand

4.21.3.17 Applicability_evaluation_by_applicability_statement_reference
The Applicability_evaluation_by_applicability_statement_reference class is a specialization of Applicability_evaluation, and makes a reference another Applicability_statement that is to be asserted as part of the Applicability_evaluation. This class enables the definition of nested applicability statements.

Applicability_evaluation_by_applicability_statement_reference associations:
− An instance of Applicability_evaluation_by_applicability_statement_reference is always associated with a specific Applicability_statement for which the nested Applicability_statement is asserted (inherited from class Applicability_evaluation)
− An instance of Applicability_evaluation_by_applicability_statement_reference is always associated with another instance of Applicability_statement which is to be asserted as part of an overall Applicability_statement.

4.21.3.18 Applicability_evaluation_by_assertion
The Applicability_evaluation_by_assertion class is a specialization of Applicability_evaluation, and identifies a value against which an applicability statement is to be asserted.

Note
The Applicability_evaluation_by_assertion class is an abstract class, ie any of its subclasses must be used for the actual representation of the value to be asserted.

Applicability_evaluation_by_assertion subclasses:
− Applicability_evaluation_by_assertion_of_condition
− Applicability_evaluation_by_assertion_of_class_instance

Applicability_evaluation_by_assertion associations:
− An instance of Applicability_evaluation_by_assertion is always associated with a specific Applicability_statement (inherited from class Applicability_evaluation).

4.21.3.19 Applicability_evaluation_by_assertion_of_condition
The Applicability_evaluation_by_assertion_of_condition class is a specialization of Applicability_evaluation_by_assertion, and identifies the condition against which an applicability statement is to be asserted.

Note
Instances of Applicability_evaluation_by_assertion_of_condition don’t have any meaning outside the context of the Applicability_statement in which it is being defined.

Applicability_evaluation_by_assertion_of_condition associations:
− An instance of Applicability_evaluation_by_assertion is always associated with a specific Applicability_statement (inherited from class Applicability_evaluation)
− An instance of Applicability_evaluation_by_assertion always refers to a value of a Condition_statement against which the applicability statement is to be asserted

4.21.3.20 Condition_statement
The Condition_statement class defines the combination of Condition_definition and a value, which needs to be true in order for the Applicability_evaluation_by_assertion_of_condition expression to be true.

Note
An instance of Condition_statement has no meaning by itself, but only in the context of an instance of Applicability_evaluation_by_assertion_of_condition.

Condition_statement association:
− An instance of Condition_statement is always related to an instance of Applicability_evaluation_by_assertion_of_condition
− An association with an instance of either Condition type or Condition_by_identifier (via the Condition_definition interface) for which a value is to be asserted
− An association with an instance of Condition_type_value that needs to be asserted, in order for the Applicability_evaluation_by_assertion_of_condition expression to be true
4.21.3.21 Applicability_evaluation_by_assertion_of_class_instance
The Applicability_evaluation_by_assertion_of_class_instance class is a specialization of Applicability_evaluation_by_assertion, and identifies a class instance against which an applicability statement is to be asserted.

**Note**
Instances of Applicability_evaluation_by_assertion_of_class_instance don’t have any meaning outside the context of the Applicability_statement in which it is being defined.

Applicability_evaluation_by_assertion_of_class_instance associations:
- An instance of Applicability_evaluation_by_assertion_of_class_instance is always associated with a specific Applicability_statement (inherited from class Applicability_evaluation)
- An instance of Applicability_evaluation_by_assertion_of_class_instance is always associated with an instance of a class, against which the applicability statement is to be asserted (via the Applicability_assert_interface)

4.21.3.22 Applicability_assert_interface
The Applicability_assert_interface is implemented by classes that have instances (values) that can be used for defining applicability statements.

Classes and interfaces that implement the Applicability_assert_interface:
- Customer
- User
- Contract
- Product_variant
- Contracted_product_variant
- Breakdown_element
- Part
- Hardware_element_realization
- Software_element_realization
- Product_usage_phase
- Product_variant_realization
- Maintenance_level_allocation_interface

Classes that implement the Applicability_assert_interface must implement the following association:
- An optional association with zero, one or many instances of Applicability_evaluation_by_assertion_of_class_instance.

4.21.3.23 Serialized_item
The Serialized_item interface is implemented by classes that can be manufactured.

Classes and interfaces that must implement the Serialized_item interface:
- Product_variant
- Part

Classes that implement the Serialized_item interface must implement the following association:
- An association with one or many instances of Block_of_serialized_items

4.21.4 UoF Applicability Statement - Additions to referenced class and interface definitions
4.21.4.1 Customer
The customer class defined in UoF Project must also implement the following additional interface:
4.21.4.2 User
The User class defined in UoF Project must also implement the following additional interface:

− Applicability_assert_interface

4.21.4.3 Contract
The Contract class defined in UoF Project must also implement the following additional interface:

− Applicability_assert_interface

4.21.4.4 Product_variant
The Product_variant class defined in UoF Project must also implement the following additional interface:

− Applicability_assert_interface
− Serialized_item

4.21.4.5 Contracted_product_variant
The Contracted_product_variant class defined in UoF Project must also implement the following additional interface:

− Applicability_assert_interface

4.21.4.6 Breakdown_element
The Breakdown_element class defined in UoF Breakdown Structure must also implement the following additional interface:

− Applicability_assert_interface

4.21.4.7 Part
The Part class defined in UoF Part must also implement the following additional interface:

− Applicability_assert_interface
− Serialized_item

4.21.4.8 Hardware_element_realization
The Hardware_element_realization class defined in UoF Breakdown Element Realization must also implement the following additional interface:

− Applicability_assert_interface

4.21.4.9 Software_element_realization
The Software_element_realization class defined in UoF Breakdown Element Realization must also implement the following additional interface:

− Applicability_assert_interface

4.21.4.10 Block_ofSerialized_items
The Block_ofSerialized_items class defined in UoF Project must also implement the following additional association:

− An optional association with an instance of Serialized_item
4.21.4.11 Product_variant_realization
The Product_variant_realization class defined in UoF Product Variant Applicability, must also implement the following additional interface:

− Applicability_assert_interface

4.21.4.12 Maintenance_level_allocation_interface
Classes that implement the Maintenance_level_allocation_interface defined in UoF Task Usage (Part 1), must also implement the following additional interface:

− Applicability_assert_interface

4.21.4.13 Substitute_part_relationship
The Substitute_part_relationship class defined in UoF Part, must also implement the following additional interface:

− Applicability_assignment

4.21.4.14 Alternate_part_relationship
The Alternate_part_relationship class defined in UoF Part, must also implement the following additional interface:

− Applicability_assignment

4.21.4.15 Hardware_part_operationalAuthorized_life
The Hardware_part_operationalAuthorized_life class defined in UoF Part, must also implement the following additional interface:

− Applicability_assignment

4.21.4.16 Parts_list_entry
The Parts_list_entry class defined in UoF Part, must also implement the following additional interface:

− Applicability_assignment

4.21.4.17 Key_performance_indicator
The Key_performance_indicator class defined in UoF LSA Candidate, must also implement the following additional interface:

− Applicability_assignment

4.21.4.18 Failure_rate_correction
The Failure_rate_correction class defined in UoF LSA Candidate, must also implement the following additional interface:

− Applicability_assignment

4.21.4.19 Mean_time_between_failure_correction
The Mean_time_between_failure_correction class defined in UoF LSA Candidate, must also implement the following additional interface:

− Applicability_assignment

4.21.4.20 LSA_failure_mode
The LSA_failure_mode class defined in UoF LSA-FMEA and Special Events, must also implement the following additional interface:

− Applicability_assignment
4.21.4.21 Failure_mode
The Failure_mode class defined in UoF LSA-FMEA and Special Events, must also implement the following additional interface:

- Applicability_assignment

4.21.4.22 Special_event_occurrence_definition
The Special_event_occurrence_definition class defined in UoF LSA-FMEA and Special Events, must also implement the following additional interface:

- Applicability_assignment

4.21.4.23 Failure_mode_effect
The Failure_mode_effect class defined in UoF LSA-FMEA and Special Events, must also implement the following additional interface:

- Applicability_assignment

4.21.4.24 Special_event_effects
The Special_event_effects class defined in UoF LSA-FMEA and Special Events, must also implement the following additional interface:

- Applicability_assignment

4.21.4.25 Task_requirement
The Task_requirement class defined in UoF LSA Candidate Task Requirement, must also implement the following additional interface:

- Applicability_assignment

4.21.4.26 Task
The Task class defined in UoF Task, must also implement the following additional interface:

- Applicability_assignment

4.21.4.27 Subtask
The Subtask class defined in UoF Task, must also implement the following additional interface:

- Applicability_assignment

4.21.4.28 Subtask_timeline
The Subtask_timeline class defined in UoF Task, must also implement the following additional interface:

- Applicability_assignment

4.21.4.29 Subtask_in_zone
The Subtask_in_zone class defined in UoF Task, must also implement the following additional interface:

- Applicability_assignment

4.21.4.30 Task_warning_caution_or_note
The Task_warning_caution_or_note class defined in UoF Task, must also implement the following additional interface:

- Applicability_assignment
4.21.4.31 Subtask_warning_caution_or_note
The Subtask_warning_caution_or_note class defined in UoF Task, must also implement the following additional interface:
− Applicability_assignment

4.21.4.32 Subtask_acceptance_parameter
The Subtask_acceptance_parameter class defined in UoF Task, must also implement the following additional interface:
− Applicability_assignment

4.21.4.33 Subtask_objective_state
The Subtask_objective_state class defined in UoF Task, must also implement the following additional interface:
− Applicability_assignment

4.21.4.34 Task_resource
The Task_resource class defined in UoF Task Resources, must also implement the following additional interface:
− Applicability_assignment

4.21.4.35 Task_resource_relationship
The Task_resource_relationship class defined in UoF Task Resources, must also implement the following additional interface:
− Applicability_assignment

4.21.4.36 Resource_realization
The Resource_realization class defined in UoF Task Resources, must also implement the following additional interface:
− Applicability_assignment

4.21.4.37 Task_personnel_resource_competence
The Task_personnel_resource_competence class defined in UoF Task Resources, must also implement the following additional interface:
− Applicability_assignment

4.21.4.38 Task_usage
The Task_usage class defined in UoF Task Usage (Part 1), must also implement the following additional interface:
− Applicability_assignment

4.21.4.39 Task_frequency
The Task_frequency class defined in UoF Task Usage (Part 1), must also implement the following additional interface:
− Applicability_assignment

4.21.4.40 Maintenance_level_allocation
The Maintenance_level_allocation class defined in UoF Task Usage (Part 1), must also implement the following additional interface:
− Applicability_assignment
4.21.4.41 Task_limit
The Task_limit class defined in UoF Task Usage (Part 1), must also implement the following additional interface:

  - Applicability_assignment

4.21.4.42 Threshold_definition
The Threshold_definition class defined in UoF Task Usage (Part 1), must also implement the following additional interface:

4.21.4.43 Applicability_assignmentSecurity_classification
The Security_classification class defined in UoF Security Classification, must also implement the following additional interface:

  - Applicability_assignment

4.21.4.44 Document_assignment
The Document_assignment class defined in UoF Document, must also implement the following additional interface:

  - Applicability_assignment

4.21.4.45 Organization_assignment
The Organization_assignment class defined in UoF Organization Assignment, must also implement the following additional interface:

  - Applicability_assignment

4.21.4.46 Classification
The Classification class defined in UoF Data Types, must also implement the following additional interface:

  - Applicability_assignment

Note
By giving the possibility to assign an Applicability_statement to an instance of Classification, there is a possibility to distinguish between different values for attributes in the data model where multiple values are allowed.

4.21.4.47 Class
The Class class defined in UoF Data Types, must also implement the following additional interface:

  - Applicability_assignment

Note
By giving the possibility to assign an Applicability_statement to an instance of Class, there is a possibility to distinguish between different values for attributes in the data model where multiple values are allowed.

4.21.4.48 Descr
The Descr class defined in UoF Data Types, must also implement the following additional interface:

  - Applicability_assignment
Note

By giving the possibility to assign an Applicability_statement to an instance of Descr, there is a possibility to distinguish between different values for attributes in the data model where multiple values are allowed.

4.21.4.49  Prp

The Prp class defined in UoF Data Types, must also implement the following additional interface:

- Applicability_assignment

Note

By giving the possibility to assign an Applicability_statement to an instance of Prp, there is a possibility to distinguish between different values for attributes in the data model where multiple values are allowed.
Chapter 20

Data exchange

Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data exchange</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>2</td>
</tr>
<tr>
<td>1 General</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Out of scope</td>
<td>4</td>
</tr>
<tr>
<td>2 Overview</td>
<td>4</td>
</tr>
<tr>
<td>2.1 S3000L Data exchanges</td>
<td>4</td>
</tr>
<tr>
<td>3 Data exchange specifications</td>
<td>4</td>
</tr>
<tr>
<td>3.1 DEX1A&amp;D Aerospace and defense business DEX for exchange of product breakdown for support</td>
<td>5</td>
</tr>
<tr>
<td>3.1.1 UoF Project in DEX1AD</td>
<td>6</td>
</tr>
<tr>
<td>3.1.2 UoF Breakdown Structure in DEX1AD</td>
<td>7</td>
</tr>
<tr>
<td>3.1.3 UoF Part in DEX1AD</td>
<td>7</td>
</tr>
<tr>
<td>3.1.4 UoF Breakdown Element Realization in DEX1AD</td>
<td>8</td>
</tr>
<tr>
<td>3.1.5 UoF Breakdown Aggregated Element in DEX1AD</td>
<td>9</td>
</tr>
<tr>
<td>3.1.6 UoF Breakdown Zone Element in DEX1AD</td>
<td>10</td>
</tr>
<tr>
<td>3.1.7 UoF Product Variant Applicability in DEX1AD</td>
<td>10</td>
</tr>
<tr>
<td>3.1.8 UoF LSA Candidate</td>
<td>11</td>
</tr>
<tr>
<td>3.1.9 UoF Security Classification in DEX1AD</td>
<td>12</td>
</tr>
<tr>
<td>3.1.10 UoF Organization Assignment in DEX1AD</td>
<td>13</td>
</tr>
<tr>
<td>3.1.11 UoF Document in DEX1AD</td>
<td>14</td>
</tr>
<tr>
<td>3.1.12 UoF Remark in DEX1AD</td>
<td>14</td>
</tr>
<tr>
<td>3.1.13 UoF Applicability Statement in DEX1AD</td>
<td>14</td>
</tr>
<tr>
<td>3.2 DEX3AD Aerospace and Defense task set</td>
<td>15</td>
</tr>
<tr>
<td>3.2.1 UoF LSA Candidate Task Requirement in DEX3AD</td>
<td>17</td>
</tr>
<tr>
<td>3.2.2 UoF Task in DEX3AD</td>
<td>18</td>
</tr>
<tr>
<td>3.2.3 UoF Task Resources in DEX3AD</td>
<td>20</td>
</tr>
<tr>
<td>3.2.4 UoF Task Usage Part 1 in DEX3AD</td>
<td>22</td>
</tr>
<tr>
<td>3.2.5 UoF Task Usage Part 2 in DEX3AD</td>
<td>24</td>
</tr>
<tr>
<td>3.2.6 UoF Security Classification in DEX3AD</td>
<td>24</td>
</tr>
<tr>
<td>3.2.7 UoF Organization Assignment in DEX3AD</td>
<td>25</td>
</tr>
<tr>
<td>3.2.8 UoF Document in DEX3AD</td>
<td>26</td>
</tr>
<tr>
<td>3.2.9 UoF Remark in DEX3AD</td>
<td>26</td>
</tr>
<tr>
<td>3.2.10 UoF Applicability Statement in DEX3AD</td>
<td>27</td>
</tr>
</tbody>
</table>

List of tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 References</td>
<td>2</td>
</tr>
</tbody>
</table>

List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 S3000L data exchanges</td>
<td>4</td>
</tr>
<tr>
<td>2 UoF Project supported by DEX1AD</td>
<td>6</td>
</tr>
<tr>
<td>3 UoF Breakdown Structure supported by DEX1AD</td>
<td>7</td>
</tr>
</tbody>
</table>
1 General

1.1 Introduction

This paragraph gives an overview of aerospace and defense data exchange specifications that supports the S3000L business process.

The Aerospace and Defense Data Exchange Specifications (AD DEX) are defined in accordance with the OASIS PLCS Data Exchange Specifications (DEX), ie each DEX has the following content:

- Terms, provides a definition of terms used in the DEX

---

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO 10303 239 PLCS</td>
<td>Data elements</td>
</tr>
<tr>
<td>DEX1AD</td>
<td>DEX1A&amp;D - Aerospace and defense business DEX for exchange of product breakdown for support (Aerospace and defense product breakdown)</td>
</tr>
<tr>
<td>DEX3AD</td>
<td>DEX3A&amp;D - Aerospace and defense business DEX for exchange of a task specification (Aerospace and defense task set)</td>
</tr>
</tbody>
</table>

www.oasis-open.org
1.2 Objective
The objective for this chapter is to define a coherent set of AD DEX that supports the S3000L LSA process and its interaction with other business processes.

1.3 Scope
The following data exchanges are in scope of this document:
− Exchange of the Product data needed for the performance of an LSA (DEX1A&D - Aerospace and defense business DEX for exchange of product breakdown for support)
− Exchange of task set data needed by eg technical publications and maintenance management (DEX3A&D - Aerospace and defense business DEX for exchange of a task specification)

1.4 Out of scope
The current set of aerospace and defense data exchange specification does not support the exchange of:
− LSA analysis activities selection
− LSA Failure Mode Effect Analysis (LSA-FMEA)

2 Overview
2.1 S3000L Data exchanges
The data exchanges that are to be supported by the current set of aerospace and defense data exchange specifications is summarized in the figure below:

![S3000L data exchanges](image)

Fig 1 S3000L data exchanges
The identified data exchange basically contains the following information:
− Product breakdown from Product development to LSA, includes information on:
  • The product to be supported,
  • One or many product variants
• One or many breakdowns of the product, including its breakdown elements (ie systems, functions, hardware, software, zones)
• Possible breakdown element realizations in terms of hardware or software parts
• Product variant applicability, and specific product variant realizations
• Key performance indicators for each LSA candidate

− Part data from material management to LSA, includes information on eg:
  • Part identifiers (eg part numbers)
  • Part name
  • Operational authorized life

− Publishing data from technical publication to LSA, includes cross reference data between S3000L tasks and S1000D data module codes in which the respective S3000L task is documented

− Task requirement from LSA to customers and Product Development, includes information on eg:
  • Task requirement specification
  • Task requirement authority
  • Task limit requirements

− Task data from LSA to customers, technical publication, materiel support, etc, includes information on eg:
  • Task identification
  • Task justification
  • Subtasks
  • Task resources

Identified data exchanges are grouped into two different aerospace and defense data exchange specifications that can be used to support one or many actual data exchanges.

The two aerospace and defense data exchange specifications that are defined in this chapter are:

− DEX1AD - DEX1A&D - Aerospace and defense business DEX for exchange of product breakdown for support
− DEX3AD - DEX3A&D - Aerospace and defense business DEX for exchange of a task specification

The content for each of those two AD DEXs are based on the unit of functionalities defined in Chap 19, Data definition.

3 Data exchange specifications
3.1 DEX1A&D Aerospace and defense business DEX for exchange of product breakdown for support

DEX1A&D - Aerospace and defense business DEX for exchange of product breakdown for support supports the following data exchanges described in Para 2:

− Product breakdown, from Product development to LSA
− Part data, from material management to LSA
− Subset of entire dataset, from external partners to LSA
− Subset of entire dataset, from LSA to external partners

The DEX1A&D basically supports the exchange of data defined in the following Unit of Functionalities (UoF) (refer to Chap 19):

− UoF Project
− UoF Breakdown Structure
A more detailed definition on the exact content in the respective unit of functionality that is supported by DEX1AD, the Aerospace and defense business DEX for exchange of product breakdown for support, is given under the respective unit of functionality figure below.

3.1.1 UoF Project in DEX1AD

For a more detailed description on the classes, attributes and associations defined in UoF Project, refer to Chap 19.

Note

The complete content of UoF Project can be exchanged using DEX1AD Aerospace and defense product breakdown for support.
3.1.2 UoF Breakdown Structure in DEX1AD

For a more detailed description on the classes, attributes and associations defined in UoF Breakdown Structure, refer to Chap 19.

Note
The complete content of UoF Breakdown Structure can be exchanged using DEX1AD Aerospace and defense product breakdown for support.

3.1.3 UoF Part in DEX1AD

For a more detailed description on the classes, attributes and associations defined in UoF Part, refer to Chap 19.
Fig 4 UoF Part supported by DEX1AD

Note
The complete content of UoF Part can be exchanged using DEX1AD Aerospace and defense product breakdown for support.

3.1.4 UoF Breakdown Element Realization in DEX1AD
For a more detailed description on the classes, attributes and associations defined in UoF Breakdown Element Realization, refer to Chap 19.
Fig 5 UoF Breakdown Element Realization supported by DEX1AD

Note
The complete content of UoF Breakdown Element Realization can be exchanged using DEX1AD Aerospace and defense product breakdown for support.

3.1.5 UoF Breakdown Aggregated Element in DEX1AD
For a more detailed description on the classes, attributes and associations defined in UoF Breakdown Aggregated Element, refer to Chap 19.
Note
The complete content of UoF Breakdown Aggregated Element can be exchanged using DEX1AD Aerospace and defense product breakdown for support.

3.1.6 UoF Breakdown Zone Element in DEX1AD
For a more detailed description on the classes, attributes and associations defined in UoF Breakdown Zone Element, refer to Chap 19.

Note
The complete content of UoF Breakdown Zone Element can be exchanged using DEX1AD Aerospace and defense product breakdown for support.

3.1.7 UoF Product Variant Applicability in DEX1AD
For a more detailed description on the classes, attributes and associations defined in UoF Product Variant Applicability, refer to Chap 19.
Note
The complete content of UoF Product Variant Applicability can be exchanged using DEX1AD Aerospace and defense product breakdown for support.

3.1.8 UoF LSA Candidate
For a more detailed description on the classes, attributes and associations defined in UoF LSA Candidate, refer to Chap 19.
3.1.9 UoF Security Classification in DEX1AD

For a more detailed description on the classes, attributes and associations defined in UoF Security Classification, refer to Chap 19.
Note

Only those security classifications that are relevant for Product breakdown can be exchanged using DEX1AD Aerospace and defense product breakdown for support.

3.1.10 UoF Organization Assignment in DEX1AD

For a more detailed description on the classes, attributes and associations defined in UoF Organization Assignment, refer to Chap 19.

Note

Only those organization assignments that are relevant for Product breakdown can be exchanged using DEX1AD Aerospace and defense product breakdown for support.
3.1.11 **UoF Document in DEX1AD**

For a more detailed description on the classes, attributes and associations defined in UoF Document, refer to [Chap 19](#).

![Diagram of UoF Document in DEX1AD](ICN-B6865-S3000L0246-002-01)

**Fig 12** UoF Document supported by DEX1AD

**Note**
- Only those document assignments that are relevant for Product breakdown can be exchanged using DEX1AD Aerospace and defense product breakdown for support.

3.1.12 **UoF Remark in DEX1AD**

For a more detailed description on the classes, attributes and associations defined in UoF Remark, refer to [Chap 19](#).
3.1.13 UoF Applicability Statement in DEX1AD

For a more detailed description on the classes, attributes and associations defined in UoF Applicability Statement, refer to Chap 19.

Note

Only those remark assignments that are relevant for Product breakdown can be exchanged using DEX1AD Aerospace and defense product breakdown for support.
Fig 14  UoF Applicability Statement supported by DEX1AD

Note
Only those applicability statement assignments that are relevant for Product breakdown can be exchanged using DEX1AD Aerospace and defense product breakdown for support.
3.2 DEX3AD Aerospace and Defense task set

The use of DEX3AD assumes that all necessary product data is exchanged through the usage of DEX1AD. The DEX3AD then adds information to the already known product with respect to how the product is to be maintained.

DEX3AD Aerospace and Defense task set supports the following data exchanges described in section 20.2:

- Publishing data, from technical publications to LSA
- Task Requirement, from LSA to customer and Product development, respectively
- Task data, from LSA to technical publications, in-service support, material management, Training and LORA, respectively
- Subset of entire dataset, from external partners to LSA
- Subset of entire dataset, from LSA to external partners

The DEX1AD DEX3AD Aerospace and Defense task set basically supports the exchange of data defined in the following units of functionality (refer to Chap 19):

- UoF LSA Candidate Task Requirement
- UoF Task
- UoF Task Resources
- UoF Task Usage (Part 1 and Part2)
- UoF Security Classification (relevant subsection of)
- UoF Organization Assignment (relevant subsection of)
- UoF Document (relevant subsection of)
- UoF Remark (relevant subsection of)
- UoF Applicability Statement (relevant subsection of)

3.2.1 UoF LSA Candidate Task Requirement in DEX3AD

For a more detailed description on the classes, attributes and associations defined in UoF LSA Candidate Task Requirement refer to Chap 19.
Fig 15  UoF LSA Candidate Task Requirement supported by DEX3AD

Note
Only those classes in the UoF LSA Task Requirement that are relevant for a Task can be exchanged using DEX3AD Aerospace and Defense task set.

Note
Detailed information about the classes shown in the shaded grey area, marked DEX1AD, is exchanged using the DEX1AD Aerospace and defense product breakdown for support. Only the attributes required for identification of the respective class are exchanged through DEX3AD.

3.2.2  UoF Task in DEX3AD
For a more detailed description on the classes, attributes and associations defined in UoF Task, refer to Chap 19.
**Fig 16** UoF Task supported by DEX3AD

**Note**
Detailed information about the classes shown in the shaded grey area, marked DEX1AD, is exchanged using the DEX1AD Aerospace and defense product breakdown for support.
Only the attributes required for identification of the respective class are exchanged through DEX3AD. These attributes are shown in UoF section 3.2.1 LSA Candidate Task Requirement in DEX3AD, above.

3.2.3 UoF Task Resources in DEX3AD

For a more detailed description on the classes, attributes and associations defined in UoF Task Resources, refer to Chap 19.
Fig 17  UoF Task Resources supported by DEX3AD

Note
Detailed information about the class in the shaded grey area, marked DEX1AD, is exchanged using the DEX1AD Aerospace and defense product breakdown for support. Only the attributes required for identification of the class is exchanged through DEX3AD. These attributes are shown in UoF section 3.2.1 LSA Candidate Task Requirement in DEX3AD, above.
3.2.4 UoF Task Usage Part 1 in DEX3AD
For a more detailed description on the classes, attributes and associations defined in UoF Task Usage Part 1, refer to Chap 19.
Fig 18  UoF Task Usage Part 1 supported by DEX3AD
Note
Detailed information about the classes in the shaded grey area, marked DEX1AD, is exchanged using the DEX1AD Aerospace and defense product breakdown for support. Only the attributes required for identification of the respective class are exchanged through DEX3AD. These attributes are shown in UoF section 3.2.1 LSA Candidate Task Requirement in DEX3AD, above.

3.2.5 UoF Task Usage Part 2 in DEX3AD
For a more detailed description on the classes, attributes and associations defined in UoF Task Usage Part 2, refer to Chap 19.

Figure 19: UoF Task Usage Part 2 supported by DEX3AD

Note
Detailed information about the classes in the shaded grey area, marked DEX1AD, is exchanged using the DEX1AD Aerospace and defense product breakdown for support. Only the attributes required for identification of the respective class are exchanged through DEX3AD. These attributes are shown in UoF section 3.2.1 LSA Candidate Task Requirement in DEX3AD, above.

3.2.6 UoF Security Classification in DEX3AD
For a more detailed description on the classes, attributes and associations defined in UoF Security Classification, refer to Chap 19.
Note
Only those security classifications that are relevant for task set can be exchanged using DEX3AD Aerospace and Defense task set. There are more classes in the UoF Security Classification that implements the Security_classification_interface, but these are out of scope for DEX3AD.

3.2.7 UoF Organization Assignment in DEX3AD
For a more detailed description on the classes, attributes and associations defined in UoF Organization Assignment, refer to Chap 19.

Note
Only those organization assignments that are relevant for task set can be exchanged using DEX3AD Aerospace and defense task set. There are more classes in the UoF Organization Assignment that implements the Organization_assignment_interface, but these are out of scope for DEX3AD.
3.2.8 UoF Document in DEX3AD

For a more detailed description on the classes, attributes and associations defined in UoF Document, refer to Chapter 19.

Fig 22 UoF Document supported by DEX3AD

Note

Only those document assignments that are relevant for task set can be exchanged using DEX3AD Aerospace and Defense task set. There are more classes in the UoF Document that implements the Organization_assignment_interface, but these are out of scope for DEX3AD.

3.2.9 UoF Remark in DEX3AD

For a more detailed description on the classes, attributes and associations defined in UoF Remark, refer to Chapter 19.
Note

Only those remark assignments that are relevant for task set can be exchanged using DEX3AD Aerospace and Defense task set. There are more classes in the UoF Remark that implements the Remark_assignment_interface, but these are out of scope for DEX3AD.

3.2.10 UoF Applicability Statement in DEX3AD

For a more detailed description on the classes, attributes and associations defined in UoF Applicability Statement, refer to Chap 19.
Fig 24  UoF Applicability Statement supported by DEX3AD
Note
Detailed information about the classes in the shaded grey area, marked DEX1AD, is exchanged using the DEX1AD Aerospace and defense product breakdown for support. Only the attributes required for identification of the respective class are exchanged through DEX3AD.

Note
Only those applicability assignments that are relevant for task set can be exchanged using DEX3AD Aerospace and Defense task set. There are more classes in the UoF Applicability Statement that implements the Applicability_assignment interface and Applicability_assert_interface, but these are out of scope for DEX3AD.
Chapter 21

Terms, abbreviations and acronyms

Table of content

<table>
<thead>
<tr>
<th>Chap 21</th>
<th>Terms, abbreviations and acronyms.................. S3000L-A-01-00-0000-00A-040A-A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 21.1</td>
<td>Terms, abbreviations and acronyms - Introduction</td>
</tr>
<tr>
<td></td>
<td>............................................................................ S3000L-A-01-01-0000-00A-040A-A</td>
</tr>
<tr>
<td>Chap 21.2</td>
<td>Terms, abbreviations and acronyms - Glossary of terms</td>
</tr>
<tr>
<td></td>
<td>........................................................................... S3000L-A-01-02-0000-00A-040A-A</td>
</tr>
<tr>
<td>Chap 21.3</td>
<td>Terms, abbreviations and acronyms - Abbreviations and acronyms</td>
</tr>
<tr>
<td></td>
<td>............................................................................ S3000L-A-01-03-0000-00A-040A-A</td>
</tr>
</tbody>
</table>
Chapter 21.1

Terms, abbreviations and acronyms - Introduction

Table of contents

<table>
<thead>
<tr>
<th>Terms, abbreviations and acronyms - Introduction</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>References ..................................................</td>
<td>1</td>
</tr>
<tr>
<td>1 General ..................................................</td>
<td>1</td>
</tr>
<tr>
<td>2 Content ..................................................</td>
<td>1</td>
</tr>
</tbody>
</table>

List of tables

| Table 1 References .................................................. | 1    |

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 21.2</td>
<td>Terms, abbreviations and acronyms - Glossary of terms</td>
</tr>
<tr>
<td>Chap 21.3</td>
<td>Terms, abbreviations and acronyms - Abbreviations and</td>
</tr>
</tbody>
</table>

1 General

This chapter gives a brief description of the main chapters that deal with information generation.

2 Content

A comprehensive terminology dictionary for terms used in this specification is given in Chap 21.2. A dictionary of abbreviations and acronyms used in this specification is presented in Chap 21.3. These include all specific S3000L definitions, abbreviations and some basic abbreviations to be used when completing Logistic Support Analysis.
Chapter 21.2

Terms, abbreviations and acronyms – Glossary of terms

Table of contents

<table>
<thead>
<tr>
<th>Terms, abbreviations and acronyms – Glossary of terms</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>References..................................................................</td>
<td>1</td>
</tr>
<tr>
<td>1 General..................................................................</td>
<td>1</td>
</tr>
<tr>
<td>2 Glossary of terms..............................................</td>
<td>1</td>
</tr>
</tbody>
</table>

List of tables

| 1 References........................................................| 1    |
| 2 Glossary of terms..............................................| 1    |

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

1 General

When there is doubt whether an abbreviation or acronym will be understood or whenever there is ample space to write in full, the term must be written out rather than abbreviated.

2 Glossary of terms

Table 2 Glossary of terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>availability</td>
<td>Availability is the measure of the degree to which an item is in an operable and ready-for-use state at the start of a mission or operation, when the mission or operation is called for at an unknown time. This is sometimes called operational readiness.</td>
</tr>
<tr>
<td>Breakdown Element Identifier</td>
<td>The Breakdown Element Identifier (BEI) is a code that identifies the breakdown structure of all system items within the LSA. The breakdown can include software items.</td>
</tr>
<tr>
<td>Breakdown Element Revision</td>
<td>The Breakdown Element Revision (BER) is a code which indicates that an alternative equipment or component occupies the same position in the breakdown and, therefore, the same installation position.</td>
</tr>
</tbody>
</table>
Term | Definition
---|---
Built In Test | Built-In tests (BIT) are implemented on items to enable them to carry out some self-testing up to a given degree. Usually, three types of built-in tests are implemented:
− power-on built-in tests (P-BIT) executed at start-up of the item
− continuous built-in tests (C-BIT) periodically and automatically executed during the operation of the item, without any intervention from the operating crew
− initiated built-in tests (I-BIT) executed upon order from the operator or from the maintenance team
Each of these types of tests detects specific categories of failures. They are characterized by:
− a detection rate that gives the percentage of functions or items whose failure is detected
− a localization or isolation rate that gives the percentage of detected failures that are associated without ambiguity to the failure of an identified replaceable unit
− a false alarm rate
Built-in tests are likely to:
− detect failures
− localize failed replaceable units
Localization may be:
− ambiguous, which means that the failure has been localized within a group of replaceable parts, not on a single replaceable part
− non-ambiguous, which means that the failed item has been identified and localized with an acceptable degree of certainty
Lifting the ambiguity is the goal of troubleshooting analysis.
In terms of warning and signals, BIT results may be signaled to the operator through a warning or alarm (which is usually the case for critical failures).
candidate item | An item which was identified to be analyzed within the LSA process.
candidate item list | List of all Candidate Items, which documents the items that are to be analyzed (documentation of the selection of analysis tasks).
chemical abstract substance identification number | Unique international identifier for a given substance.
commercial off-the-shelf | Software or hardware, generally technology products, that is ready-made and available for sale, lease, or license to the general public.
common cause | Some failures may lead to several malfunctions. For instance, the failure of a power supply leads to malfunction of all its supplied items. This type of failure with multiple impacts is called a common cause.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>corrective maintenance</td>
<td>All maintenance activities, which are carried out to reset a faulty item to full functionality.</td>
</tr>
<tr>
<td>customer requirements</td>
<td>A document containing all the logistic requirements coming from customer. This document should be available for the Guidance Conference.</td>
</tr>
<tr>
<td>damage</td>
<td>A loss or reduction of functionality, excluding inherent failure (intrinsic reliabilities). Normally a maintenance task will be required. Damage can be grouped into &quot;damage families&quot;, e.g., concerning structures; typical damage can be identified like scratches, dents or cracks. These damage families are typical candidates for a standard repair procedure.</td>
</tr>
<tr>
<td>dangerous/toxic substance</td>
<td>Substance which harms human beings and environment (animals, plants, etc.).</td>
</tr>
<tr>
<td>data element list</td>
<td>List of selected data elements or an output of a data element tailoring process. This list can contain additional data elements required for a special project, which are not predefined in any standard.</td>
</tr>
<tr>
<td>defect</td>
<td>Any non-conformance of an item with its specified requirements is a defect. Note that a defect does not necessarily result in a failure of the item.</td>
</tr>
<tr>
<td>definition files</td>
<td>Definition files, such as technical descriptions, interface definitions, wiring diagrams, description of sockets bounding, etc. may be useful if localization remains ambiguous after all simple means of troubleshooting have been used and when it becomes necessary to make functional or electrical tests to help localize the failed item.</td>
</tr>
<tr>
<td>demilitarization</td>
<td>Operation which makes a defense Product unavailable for its intended military use.</td>
</tr>
<tr>
<td>detection</td>
<td>Failure detection warns the operator that a failure has occurred.</td>
</tr>
<tr>
<td>direct replaceable</td>
<td>Item that is directly replaceable on the product. No functional testing is required.</td>
</tr>
<tr>
<td>deviation</td>
<td>Authorized approval to depart from a particular requirement of a product/equipment approved configuration documentation for a specified period of time. This allows the acceptance of an equipment/product which departs from the particular requirement, but is considered as suitable for use 'as is' or after repair by an approved method.</td>
</tr>
<tr>
<td>disposal phase</td>
<td>Last phase of a Product Life Cycle, where the product is phased out of use.</td>
</tr>
<tr>
<td>disposal process</td>
<td>Set of tasks to be performed on a product at the end of its intended use.</td>
</tr>
<tr>
<td>external cause</td>
<td>A cause is said to be external when an event independent of product usage occurs e.g., bird-strike.</td>
</tr>
<tr>
<td>failure</td>
<td>Unacceptable reduction of functionality of an item where the item cannot continue in its intended use. The failure occurs inherently during proper usage of the item.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>failure cause</td>
<td>A failure cause is any circumstance during design, manufacture or use which led to the failure.</td>
</tr>
<tr>
<td>failure mode</td>
<td>A failure mode is a predicted or observed physical, mechanical, thermal or other process which leads to a failure. The result of this process is stated in relation to the operating conditions at the time of the failure.</td>
</tr>
<tr>
<td>Failure rate</td>
<td>The failure rate is the probability of failure per unit of time of items in operation.</td>
</tr>
<tr>
<td>Failure Modes and Effects Analysis (FMEA)</td>
<td>A Failure Modes and Effects Analysis (FMEA) is a procedure for analysis of potential failure modes within a system for classification by severity or determination of the effect of failures on the system.</td>
</tr>
<tr>
<td>Failure Mode, Effects, and Criticality Analysis (FMECA)</td>
<td>Failure Mode, Effects, and Criticality Analysis (FMECA) is an extension of Failure Mode and Effects Analysis (FMEA). In addition to the basic FMEA, it includes a criticality analysis, which is used to chart the probability of failure modes against the severity of their consequences. The result highlights failure modes with relatively high probability and severity of consequences, allowing remedial effort to be directed where it will produce the greatest value.</td>
</tr>
<tr>
<td>field loadable software</td>
<td>Software that can be installed to one or several equipments on a system/product without need to dismount the target from its installation location.</td>
</tr>
<tr>
<td>firmware</td>
<td>Software that can be loaded into a LRU or SRU but requires the target to be dismounted from its installation on the operational system and requiring the replacement of a component.</td>
</tr>
<tr>
<td>Fit form and function</td>
<td>Where the physical interface, physical parameters that uniquely characterize or the actions of a product/equipment remain unchanged.</td>
</tr>
<tr>
<td>function breakdown</td>
<td>A simple functional breakdown can start on top level of the product as the root of the breakdown tree. The different functions of the product should be documented from the main functions down to the sub-functions, etc, to the required depth. Within the pure functional breakdown, no hardware elements should be found.</td>
</tr>
<tr>
<td>functionality analysis</td>
<td>Functional analysis describes and links functions performed by the system. Each function of the system may be characterized by the flows or parameters:</td>
</tr>
</tbody>
</table>
|                                          | − it takes as inputs
|                                          | − it requires as part of operating context
|                                          | − it provides as outputs
<p>|                                          | The providing of the output flows justifies the presence of the items on board. Failure prevention, detection and correction all aim to the continuity or the restoration of these functions. Each function may also be associated with a list of the physical items which realize the function or contribute to it. This helps in identifying the functions impacted by failure of given equipment. |</p>
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>functional symptom</td>
<td>A functional symptom characterizes a failure detected by functional check and/or by loss or degradation of an operational function. It is detectable when the system is currently operated or checked via a functional check.</td>
</tr>
<tr>
<td>guidance conference</td>
<td>Conference to establish the LSA business process (breath and depth of analysis) with contractor and customer members, where the basic and binding methods and procedures should be agreed.</td>
</tr>
<tr>
<td>guidance conference document</td>
<td>A document with contractual character, where agreements between the customer and the contractor are documented and which is signed by both sides. This document is an output of the Guidance Conference.</td>
</tr>
<tr>
<td>integrated logistic support</td>
<td>The management process which facilitates development and integration between individual logistic support elements to specify, design, develop, acquire, test, field, and support technical systems.</td>
</tr>
<tr>
<td>internal cause</td>
<td>A cause is said internal when it comes from product usage by itself eg excessive vibration.</td>
</tr>
<tr>
<td>labor time</td>
<td>Summarized duration of personnel work. The duration should be addressed against the subtasks. If more than one person is working parallel on a subtask, the Labor time for each different skill level must be summarized.</td>
</tr>
<tr>
<td>Level Of Repair Analysis</td>
<td>Level Of Repair Analysis (LORA) is a prescribed procedure for logistics planning. LORA is performed to determine the best, most efficient location where an item can be repaired.</td>
</tr>
<tr>
<td>localization</td>
<td>Failure localization indicates which item or group of items has failed. This localization is generally a complement to failure detection</td>
</tr>
<tr>
<td>Logistic Support Analysis (LSA)</td>
<td>The selective application of scientific and engineering efforts undertaken during the development process and continuing throughout the complete life cycle, as part of the system engineering and design process, to assist in complying with supportability and other Integrated Logistic Support activities.</td>
</tr>
<tr>
<td>LSA database</td>
<td>A database, where the relevant logistic data is collected. Other terms can be LSA Record (LSAR) or support engineering database.</td>
</tr>
<tr>
<td>life cycle</td>
<td>Total product lifespan since its initial feasibility phase, until its service withdrawal and total dismantling of the park.</td>
</tr>
<tr>
<td>Life Cycle Costs</td>
<td>Life Cycle Cost (LCC) is the cumulative cost of a product over its life cycle from entry into service to disposal as determined by a process of economic analysis that allows for the assessment of the total cost of acquisition, ownership and disposal of the product.</td>
</tr>
<tr>
<td>maintainability</td>
<td>The measure of the ability of an item to be retained in or restored to a specified condition, when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance and repair.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maintenance Concept</td>
<td>A statement of maintenance considerations, constraints and strategy for the operational support that governs the maintenance levels and type of maintenance activities to be carried out for the system/equipment under analysis.</td>
</tr>
<tr>
<td>Maintenance Plan</td>
<td>The purpose of the maintenance plan is to ensure that the equipment/product can be maintained effectively and economically at the desired level of readiness after it is placed in operational use. The Maintenance plan specifies when, where, and how maintenance tasks will be performed on the equipment/product, including both:</td>
</tr>
<tr>
<td></td>
<td>• Preventive maintenance</td>
</tr>
<tr>
<td></td>
<td>• Corrective maintenance</td>
</tr>
<tr>
<td>maintenance task analysis</td>
<td>The Maintenance Task Analysis is a detailed analysis performed for each of the logistic related operational or maintenance tasks and will identify the support resources required to implement effective corrective and preventative maintenance tasks for a system and/or equipment. These maintenance tasks are identified in the LSA process.</td>
</tr>
<tr>
<td>maintenance relevant item</td>
<td>A Maintenance Relevant Item is an item that can be repaired or replaced in case of a failure or damage. Normally this item is automatically a potential LSA candidate, but not automatically an MSI.</td>
</tr>
<tr>
<td>Maintenance Steering Group 3 (MSG-3)</td>
<td>A designation for an established procedure of scheduled maintenance analysis. The ATA MSG-3 publication outlines a decision-logic process for determining initial scheduled maintenance requirements for new aircraft and/or power plants. This document presents a means for developing maintenance tasks and intervals acceptable to regulatory authorities, operators and manufacturers.</td>
</tr>
<tr>
<td>maintenance significant item</td>
<td>A Maintenance Significant Item (MSI) is an item that was identified by any selection process as the result of a Scheduled Maintenance Analysis procedure such as S4000M, MSG-3 or RCM. For this item, a Scheduled Maintenance Analysis (SMA) will be performed (eg a system and power plant analysis as described in S4000M). A Maintenance Significant Item can become a LSA candidate if the SMA identifies a scheduled task. This task will be documented in the LSA database.</td>
</tr>
<tr>
<td>malfunction analysis</td>
<td>Malfunctioning analysis aims to define what to do when things go wrong. It may cover two types of corrective actions:</td>
</tr>
<tr>
<td></td>
<td>- actions to be taken by the operating crew when a failure occurs during operation,</td>
</tr>
<tr>
<td></td>
<td>- actions to be taken as part of the corrective maintenance.</td>
</tr>
<tr>
<td></td>
<td>Only the second type of actions falls into the scope of this specification.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mean elapsed time</td>
<td>Duration of the entire task. The duration should be addressed against the subtasks. The duration of the whole task can be calculated as the sum of times coming from the subtasks and from the referenced supporting tasks.</td>
</tr>
<tr>
<td>Mean time between failures</td>
<td>Mean time between failure is the predicted elapsed time between inherent failures of a system during operation. MTBF can be calculated as the arithmetic mean (average) time between failures of a system. The MTBF is typically part of a model that assumes the failed system is immediately repaired (zero elapsed time), as a part of a renewal process. This is in contrast to the Mean Time To Failure (MTTF), which measures average time between failure with the modeling assumption that the failed system is not repaired.</td>
</tr>
<tr>
<td>Model Identifier</td>
<td>The Model Identifier is a code, which uniquely identifies the product variant. It is recommended to use the model identifier in conjunction with other identifiers within the entire ILS process.</td>
</tr>
<tr>
<td>operational requirements document</td>
<td>Document which defines the operational requirements and needs in a qualitative way for the end item platform. This document should be available for the Guidance Conference.</td>
</tr>
<tr>
<td>physical breakdown</td>
<td>A top down representation of hardware and software of a product based on the engineering design model/drawings.</td>
</tr>
<tr>
<td>physical symptom</td>
<td>A physical symptom characterizes a failure detected by visual inspection, measurement of a wear-out parameter, material degradation. It is detectable or measurable when the system is currently operated or if it is undergoing inspection or maintenance.</td>
</tr>
<tr>
<td>preventive maintenance</td>
<td>Maintenance activities to prevent the occurrence of critical failures or damages in conjunction with safety, economical or ecological aspects. The Preventive Maintenance also includes activities after special events where these events, chronological intervals or other regular threshold cannot be defined.</td>
</tr>
<tr>
<td>product</td>
<td>The Product is a final combination of systems, subsystems, component parts/materials, etc. such as an aircraft, a ship, vehicle, or a complex technical system. The product always represents the top level of any hierarchical breakdown.</td>
</tr>
<tr>
<td>rectifying task</td>
<td>A maintenance activity, which resolves an event such as failures, damages, special events or thresholds. A rectifying task contains subtasks in terms of referenced supporting tasks and/or definite working steps.</td>
</tr>
<tr>
<td>reliability</td>
<td>The duration or probability of failure free performance of a product or system under stated conditions, or the probability that an item can perform its intended function for a specified interval under stated conditions, is a prime driver of support resources.</td>
</tr>
<tr>
<td>reliability centered maintenance</td>
<td>A disciplined logic or methodology used to identify scheduled maintenance tasks to maintain the inherent reliability of equipment at a minimum expenditure of resources.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>scheduled maintenance</td>
<td>Maintenance activities to prevent the occurrence of critical failures or damages in conjunction with safety, economical or ecological aspects. These maintenance tasks are defined with a corresponding interval or threshold eg after certain time, cycles, rounds, distance. Scheduled maintenance is a subset of the preventive maintenance.</td>
</tr>
<tr>
<td>SMA Candidate</td>
<td>Element of the breakdown, which is identified as a potential candidate for which a scheduled or preventive maintenance task could be necessary.</td>
</tr>
<tr>
<td>Scheduled Maintenance Analysis (SMA)</td>
<td>The analysis task for the identification of preventive and scheduled maintenance. Different methods are available to perform this analysis task such as S4000M, MSG-3 or RCM.</td>
</tr>
<tr>
<td>Shop Loadable Software</td>
<td>Software that can be loaded into a LRU but that requires the target LRU to be dismounted from its installation in the system where it is located.</td>
</tr>
<tr>
<td>Shop Replaceable Unit (SRU)</td>
<td>An item, which can only be changed at shop/rear level (cannot be removed and installed at the organizational maintenance area).</td>
</tr>
<tr>
<td>signature</td>
<td>A list of physical symptoms, functional symptoms and BIT results that help detect and localize the failure.</td>
</tr>
<tr>
<td>Software Support Analysis (SSA)</td>
<td>Methodology to analyze software with the purpose to supply the customer with all necessary information to establish a cost effective SSC. This includes all equipment, tools, personnel, documentation, infrastructure and required skill and training respectively.</td>
</tr>
<tr>
<td>special event</td>
<td>A special event is something that may occur during the life of the product but cannot be considered as a normal way of operation. It can be due either to external causes, eg meteorological phenomenon for example or to out of bound use such as over G maneuver.</td>
</tr>
<tr>
<td>specific test equipment</td>
<td>Whenever built-in-tests are unable to detect or localize failures, some specific test equipment may be required. This specific test equipment may be:</td>
</tr>
<tr>
<td></td>
<td>− external measurement or test items</td>
</tr>
<tr>
<td></td>
<td>− external means used to simulate a function of the operating system</td>
</tr>
<tr>
<td>structural detail</td>
<td>A specific area of the Structure Significant Item which was identified by any selection process from a SMA procedure such as S4000M or MSG-3. For this detail, a SMA will be performed.</td>
</tr>
<tr>
<td>structural item</td>
<td>A Structural Item is a part of the system bodywork. It can be an LSA candidate as well as a Structure Significant Item.</td>
</tr>
<tr>
<td>structure significant item</td>
<td>An item which is identified by the selection process from a Scheduled Maintenance Analysis procedure such as S4000M or MSG-3.</td>
</tr>
<tr>
<td>substances cartography</td>
<td>Characterization (identification, risk, quantity etc) and location of all the tracked substances within a Product.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Supply Chain</td>
<td>A supply chain consists of all organizations involved, directly or indirectly, in fulfilling a customer support requirements.</td>
</tr>
<tr>
<td>supply chain management</td>
<td>The process of planning, implementing and controlling the operations of the supply chain as efficiently as possible.</td>
</tr>
<tr>
<td>supporting task</td>
<td>A supporting task is a part of a complete maintenance activity. As a standalone task it cannot rectify an event like failures, damages, special events or thresholds. A supporting task contains subtasks in terms of working steps.</td>
</tr>
</tbody>
</table>
| testability               | Performances of Built-in-Tests (BIT) developed on some items of the system are often assessed using FMEA terms and results, but this is not mandatory, especially if the item has been previously developed for another program. Whatever their elaboration method, BIT may be characterized by:  
  - a list of functions or sub-functions that are tested  
  - a detection rate that gives the percentage of functions or items whose failure is detected  
  - a localization or isolation rate that gives the percentage of detected failures that are associated without ambiguity to the failure of an identified item or component  
  - a false alarm rate  
These analyses and their justification should be used for malfunctioning analysis and for LSA FMEA, especially because they provide:  
  - a list of failure reports (from BIT) giving failure codes or any other message that indicates which item or group of items is liable to have failed, therefore restraining greatly the number of items to be investigated as part of the troubleshooting process  
Warnings sent to the operating crew or to the maintenance team to indicate detected failures. These warnings should be used as first-hand symptoms rather than any other clue on the item itself |
| Training Needs Analysis (TNA) | Analysis task to identify the requirements for training concerning the operational and maintenance activities.                                    |
| troubleshooting           | Troubleshooting consists of localizing failed replaceable units when this is not obvious or previously done by other means eg built-in test. Troubleshooting is carried out after a failure has been detected. |
| unified identifier        | Designation for an identifier (ID), which can be used across all logistic disciplines in the ILS process for a definite identification of any item. |
| Usable on Code (UoC)      | A model identifier that indicates the configuration of a product on which the item under analysis is used.                                  |
Chapter 21.3

Terms, abbreviations and acronyms - Abbreviations and acronyms

Table of contents

<table>
<thead>
<tr>
<th>Terms, abbreviations and acronyms - Abbreviations and acronyms</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>References</td>
<td>1</td>
</tr>
<tr>
<td>1 General</td>
<td>1</td>
</tr>
<tr>
<td>2 Word combination - Acronym</td>
<td>1</td>
</tr>
<tr>
<td>3 Tense and number</td>
<td>1</td>
</tr>
<tr>
<td>4 Abbreviation and acronym list</td>
<td>2</td>
</tr>
</tbody>
</table>

List of tables

1 References                                                                                         | 1 |
2 Abbreviations and acronyms                                                                           | 2 |

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 General

When there is doubt that an abbreviation or acronym will be understood or whenever there is ample space to write in full, the term must be written out rather than abbreviated. Abbreviations and acronyms listed in Table 2 reflect their use in S3000L and not in other documents.

2 Word combination - Acronym

Abbreviations for word combinations, acronyms, must be used as such and not separated for use singly, unless authorized singly.

Single abbreviations can be combined when necessary if there is no abbreviation listed for the combination.

3 Tense and number

The same abbreviation must be used for all tenses, possessive cases, singular and plural forms of a given word.
# Abbreviation and acronym list

<table>
<thead>
<tr>
<th>Abbreviation /Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIA</td>
<td>Aerospace Industries Association of America</td>
</tr>
<tr>
<td>AOR</td>
<td>Annual Operating Rate</td>
</tr>
<tr>
<td>ASD</td>
<td>AeroSpace and Defense Industries Association of Europe</td>
</tr>
<tr>
<td>ATA</td>
<td>Air Transport Association of America</td>
</tr>
<tr>
<td>BE</td>
<td>Breakdown Element</td>
</tr>
<tr>
<td>BEI</td>
<td>Breakdown Element Identifier</td>
</tr>
<tr>
<td>BER</td>
<td>Breakdown Element Revision</td>
</tr>
<tr>
<td>BIT</td>
<td>Built in Test</td>
</tr>
<tr>
<td>BITE</td>
<td>Built in Test Equipment</td>
</tr>
<tr>
<td>CBS</td>
<td>Cost Breakdown Structure</td>
</tr>
<tr>
<td>CE</td>
<td>Communauté Européenne</td>
</tr>
<tr>
<td>CEE</td>
<td>Communauté Economique Européenne</td>
</tr>
<tr>
<td>Chap</td>
<td>Chapter</td>
</tr>
<tr>
<td>CI</td>
<td>Candidate Item</td>
</tr>
<tr>
<td>CIL</td>
<td>Candidate Item List</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off The Shelf</td>
</tr>
<tr>
<td>CP</td>
<td>Change Proposal</td>
</tr>
<tr>
<td>CRD</td>
<td>Customer Requirements Document</td>
</tr>
<tr>
<td>DA</td>
<td>Design Authority</td>
</tr>
<tr>
<td>DEL</td>
<td>Data Element List</td>
</tr>
<tr>
<td>Descr</td>
<td>Description</td>
</tr>
<tr>
<td>DEX</td>
<td>Data Exchange Specification</td>
</tr>
<tr>
<td>DM</td>
<td>Data Module</td>
</tr>
<tr>
<td>DMC</td>
<td>Data Module Code</td>
</tr>
<tr>
<td>eg</td>
<td>for example</td>
</tr>
<tr>
<td>ELORA</td>
<td>Economical Level of Repair Analysis</td>
</tr>
<tr>
<td>Fig</td>
<td>Figure</td>
</tr>
<tr>
<td>FLS</td>
<td>Field Loadable Software</td>
</tr>
<tr>
<td>Abbreviation / Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>FMEA</td>
<td>Failure Mode and Effects Analysis</td>
</tr>
<tr>
<td>FMECA</td>
<td>Failure Mode and Effects Criticality Analysis</td>
</tr>
<tr>
<td>FMR</td>
<td>Failure Mode Ratio</td>
</tr>
<tr>
<td>FSN</td>
<td>Fleet Serial Number</td>
</tr>
<tr>
<td>GC</td>
<td>Guidance Conference</td>
</tr>
<tr>
<td>GCD</td>
<td>Guidance Conference Document</td>
</tr>
<tr>
<td>GEIA</td>
<td>Government Electronic and Information Technology Association</td>
</tr>
<tr>
<td>ICN</td>
<td>Information Control Number</td>
</tr>
<tr>
<td>ID</td>
<td>Identifier</td>
</tr>
<tr>
<td>ie</td>
<td>that is</td>
</tr>
<tr>
<td>ILS</td>
<td>Integrated Logistic Support</td>
</tr>
<tr>
<td>ISO</td>
<td>International Standards Organization</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>IUA</td>
<td>Item under Analysis</td>
</tr>
<tr>
<td>kg</td>
<td>kilogram</td>
</tr>
<tr>
<td>lb</td>
<td>pound</td>
</tr>
<tr>
<td>LCC</td>
<td>Life Cycle Costs</td>
</tr>
<tr>
<td>LCMP</td>
<td>LSA Configuration Management Plan</td>
</tr>
<tr>
<td>LCN</td>
<td>Logistic Control Number</td>
</tr>
<tr>
<td>LORA</td>
<td>Level of Repair Analysis</td>
</tr>
<tr>
<td>LRU</td>
<td>Line Replaceable Unit</td>
</tr>
<tr>
<td>LSA</td>
<td>Logistic Support Analysis</td>
</tr>
<tr>
<td>LSA RC</td>
<td>Logistic Support Analysis Review Conference</td>
</tr>
<tr>
<td>$\bar{M}$</td>
<td>Mean active maintenance action time</td>
</tr>
<tr>
<td>MC</td>
<td>Maintenance Concept</td>
</tr>
<tr>
<td>MDT</td>
<td>Maintenance Down Time</td>
</tr>
<tr>
<td>MET</td>
<td>Mean Elapsed Time</td>
</tr>
<tr>
<td>MIL-STD</td>
<td>Military Standard (US DoD)</td>
</tr>
<tr>
<td>ML</td>
<td>Maintenance Level</td>
</tr>
<tr>
<td>MoD</td>
<td>Ministry of Defence</td>
</tr>
<tr>
<td>MSDS</td>
<td>Material Safety Data Sheet</td>
</tr>
<tr>
<td>Abbreviation / Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>MSG</td>
<td>Maintenance Steering Group</td>
</tr>
<tr>
<td>MSI</td>
<td>Maintenance Significant Item</td>
</tr>
<tr>
<td>MSN</td>
<td>Manufacturer Serial Number</td>
</tr>
<tr>
<td>MTA</td>
<td>Maintenance Task Analysis</td>
</tr>
<tr>
<td>MTBF</td>
<td>Mean Time Between Failures</td>
</tr>
<tr>
<td>MTBM</td>
<td>Mean Time Between Maintenance</td>
</tr>
<tr>
<td>NATO</td>
<td>North Atlantic Treaty Organization</td>
</tr>
<tr>
<td>O&amp;S</td>
<td>Operating and Support</td>
</tr>
<tr>
<td>OASIS</td>
<td>Organisation for the Advancement of Structural Information Standards</td>
</tr>
<tr>
<td>OEM</td>
<td>Original Equipment Manufacturer</td>
</tr>
<tr>
<td>OM</td>
<td>Obsolescence management</td>
</tr>
<tr>
<td>ORD</td>
<td>Operational Requirements Document</td>
</tr>
<tr>
<td>para</td>
<td>paragraph</td>
</tr>
<tr>
<td>PBS</td>
<td>Product Breakdown Structure</td>
</tr>
<tr>
<td>PDF</td>
<td>Product Disposal File</td>
</tr>
<tr>
<td>PDM</td>
<td>Product Data Management</td>
</tr>
<tr>
<td>PE</td>
<td>Periodical</td>
</tr>
<tr>
<td>PHST</td>
<td>Packaging, Handling, Storage and Transportation</td>
</tr>
<tr>
<td>PLCS</td>
<td>Product Life Cycle Support</td>
</tr>
<tr>
<td>P/N</td>
<td>Part number</td>
</tr>
<tr>
<td>PNR</td>
<td>Part number</td>
</tr>
<tr>
<td>PO</td>
<td>Perform once</td>
</tr>
<tr>
<td>PP</td>
<td>Program Plan</td>
</tr>
<tr>
<td>RCM</td>
<td>Reliability Centered Maintenance</td>
</tr>
<tr>
<td>RDL</td>
<td>Reference Data Library</td>
</tr>
<tr>
<td>RFC</td>
<td>Request for clarification</td>
</tr>
<tr>
<td>SB</td>
<td>Service Bulletin</td>
</tr>
<tr>
<td>SI</td>
<td>Structural Item</td>
</tr>
<tr>
<td>SMA</td>
<td>Scheduled Maintenance Analysis</td>
</tr>
<tr>
<td>SRU</td>
<td>Shop Replaceable Unit</td>
</tr>
<tr>
<td>SSA</td>
<td>Software Support Analysis</td>
</tr>
<tr>
<td>SSC</td>
<td>Software Support Concept</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>ST</td>
<td>SubTask</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>TNA</td>
<td>Training Needs Analysis</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language™</td>
</tr>
<tr>
<td>UoF</td>
<td>Unit of Functionality</td>
</tr>
<tr>
<td>WBS</td>
<td>Work Breakdown Structure</td>
</tr>
</tbody>
</table>
Chapter 22

Data element list

Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data element list</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>1</td>
</tr>
<tr>
<td>1 General</td>
<td>1</td>
</tr>
</tbody>
</table>

List of tables

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>References ................................................................. 1</td>
</tr>
<tr>
<td>2</td>
<td>Data elements in alphabetical order ......................... 2</td>
</tr>
<tr>
<td>3</td>
<td>Attributes of the S3000L data types ............................ 55</td>
</tr>
</tbody>
</table>

References

Table 1 References

<table>
<thead>
<tr>
<th>Chap No./Document No.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 19</td>
<td>Data elements</td>
</tr>
<tr>
<td>Chap 20</td>
<td>Data exchange</td>
</tr>
<tr>
<td>S1000D</td>
<td>International specification for technical publications using a common source database</td>
</tr>
<tr>
<td>S2000M</td>
<td>International specification for material management</td>
</tr>
<tr>
<td>S4000M</td>
<td>International procedure handbook for the development of scheduled maintenance programs for military aircrafts</td>
</tr>
</tbody>
</table>

1 General

This chapter defines all the data elements that are used as attributes in the S3000L data model (Chap 19), and in the S3000L data exchange specifications (Chap 20).

The data element list in Table 2 defines all business related data elements. The table is organized alphabetically by the data element name, and contains:

- Data element name
- Data element data type (refer to Chap 19 on more details on data types used in S3000L)
- Data element definition, contains a textual definition and a list of valid values
- Class name, identifies Classes in the S3000L data model where the data element is used as an attribute (Chap 19)
- Unit of Functionality, identifies the section in Chap 19 where the Class is defined

The attribute list in Table 3 defines all the data elements that are used to define the S3000L data types.
### Table 2: Data elements in alphabetical order

<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual_number_of_lines_of_code</td>
<td>Prp</td>
<td>Actual number of lines of executable code in a software part. The following Unit of Measures is recommended to be used to enable integration with other ASD specifications: - 'Lines_of_software_code'</td>
<td>Software_part</td>
<td>Part</td>
</tr>
<tr>
<td>Actual_software_complexity_code</td>
<td>Class</td>
<td>Code for describing software part complexity. Valid classes are to be determined by the project.</td>
<td>Software_part</td>
<td>Part</td>
</tr>
<tr>
<td>Actual_software_size</td>
<td>Prp</td>
<td>Actual size of a software part. The following Unit of Measures is recommended to be used to enable integration with other ASD specifications: - ‘Kilobyte’ - ‘Megabyte’ - ‘Gigabyte’ - ‘Terabyte’</td>
<td>Software_part</td>
<td>Part</td>
</tr>
<tr>
<td>Aggregated_element_description</td>
<td>Descr</td>
<td>Description of an aggregated breakdown element.</td>
<td>Aggregated_element_revision</td>
<td>Breakdown Aggregated Element</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Aggregated_element_type</td>
<td>Class</td>
<td>Indicates further specialization of an aggregated breakdown element.</td>
<td>Aggregated_element</td>
<td>Breakdown Aggregated Element</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes is recommended to be used to enable integration with</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘System_breakdown_element’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Functional_breakdown_element’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Family_breakdown_element’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Slot_breakdown_element’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicability_end_date</td>
<td>Date</td>
<td>The date that defines the upper bound of the interval for a dated applicability.</td>
<td>Dated_applicability_statement</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td>Applicability_start_date</td>
<td>Date</td>
<td>The date that defines the lower bound of the interval for a dated applicability.</td>
<td>Dated_applicability_statement</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td>Applicability_statement_description</td>
<td>Descr</td>
<td>Human readable applicability information. Provides a textual description of</td>
<td>Applicability_statement</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>an applicability statement and its conditions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicability_statement_identifier</td>
<td>Id</td>
<td>Identification of an applicability statement.</td>
<td>Applicability_statement</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breakdown_element_identifier</td>
<td>Id</td>
<td>Identification of an element used within a breakdown of a defined product.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Note 1: Used to establish a hierarchical structure of the technical system (known as LCN/ALCN within GEIA-STD-0007).</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following Identifier_class is recommended to be used for Breakdown_element_identifier, to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'ASD_system_or_hardware_identification_code' (also known as AECMA-code or SNS in S1000D)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breakdown_element_name</td>
<td>Id</td>
<td>The name by which a breakdown element is known.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following Identifier_class is recommended to be used for Breakdown_element_name, to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'ASD_technical_name' ('techname' in S1000D).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Breakdown_element_relationship_type</td>
<td>Class</td>
<td>Definition of the type of relationship that is being established between two breakdown elements.</td>
<td>Breakdown_element_relationship</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Note: The related breakdown elements does not need to be used in the same breakdown, i.e. it can be used to establish the relationship between a breakdown element in a functional breakdown and a breakdown element in a physical breakdown.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are recommended to be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Alternative_breakdown_element'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Functional_and_physical_breakdown_element_relationship'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Access_point'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breakdown_element_revision_creation_date</td>
<td>Date</td>
<td>Date when the breakdown element revision was created.</td>
<td>Breakdown_element_revision</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td>Breakdown_element_revision_identifier</td>
<td>Id</td>
<td>Identification of a specific revision of a breakdown element.</td>
<td>Breakdown_element_revision</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Note: Revision identifier is used to identify design iterations and not variants.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breakdown_element_revision_status</td>
<td>Class</td>
<td>Indication of the design progression of a breakdown element used in a breakdown of a product.</td>
<td>Breakdown_element_revision</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid classes are to be determined by the project.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breakdown_revision_creation_date</td>
<td>Date</td>
<td>Date when the breakdown revision was created.</td>
<td>Breakdown_revision</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Breakdown_revision_identifier</td>
<td>Id</td>
<td>Identification of a specific revision of a breakdown.</td>
<td>Breakdown_revision</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Note: Revision identification is used to document design iterations and not breakdown variants.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breakdown_revision_status</td>
<td>Class</td>
<td>Indication of the design progression of a complete breakdown of a product.</td>
<td>Breakdown_revision</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid classes are to be determined by the project.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breakdown_type</td>
<td>Class</td>
<td>The type of breakdown being defined.</td>
<td>Breakdown</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are recommended to be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Physical_breakdown'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Functional_breakdown'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'System_breakdown'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'ASD_system_hardware_breakdown'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Zonal_breakdown'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Family_tree_breakdown'.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Candidate_item_analysis_activity_date</td>
<td>Date</td>
<td>Date for the latest update of the candidate item analysis activity information.</td>
<td>Candidate_item_analysis_activity</td>
<td>LSA Candidate Analysis Activity</td>
</tr>
<tr>
<td>Candidate_item_analysis_activity_indicator</td>
<td>Class</td>
<td>Definition of whether the candidate item analysis activity is selected for a specific LSA candidate.</td>
<td>Candidate_item_analysis_activity</td>
<td>LSA Candidate Analysis Activity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are recommended to be used:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Selected'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Candidate_item_analysis_activity_rationale</td>
<td>Descr</td>
<td>Description of the reason for the selection / non-selection of the candidate item analysis activity.</td>
<td>Candidate_item_analysis_activity</td>
<td>LSA Candidate Analysis Activity</td>
</tr>
<tr>
<td>Candidate_item_analysis_activity_status</td>
<td>Class</td>
<td>Indication of the progression of the candidate item analysis activity.</td>
<td>Candidate_item_analysis_activity</td>
<td>LSA Candidate Analysis Activity</td>
</tr>
<tr>
<td>Change_request_description</td>
<td>Descr</td>
<td>Description of a desired product design change.</td>
<td>Change_request</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Change_request_identifier</td>
<td>Id</td>
<td>Identification of a desired product design change, eg due to a task requirement.</td>
<td>Change_request</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Circuit_breaker_identifier</td>
<td>Id</td>
<td>Identification of a circuit breaker that is installed on the end item.</td>
<td>Circuit_breaker</td>
<td>Task</td>
</tr>
<tr>
<td>Circuit_breaker_name</td>
<td>Id</td>
<td>The name by which a circuit breaker is known.</td>
<td>Circuit_breaker</td>
<td>Task</td>
</tr>
<tr>
<td>Circuit_breaker_type</td>
<td>Class</td>
<td>Definition of the type of circuit breaker being identified.</td>
<td>Circuit_breaker</td>
<td>Task</td>
</tr>
</tbody>
</table>

- ‘Rejected’
- ‘Open (not decided)’

The following classes are recommended to be used:
- ‘Not_started’
- ‘Ongoing’
- ‘Completed’

The following classes is recommended to be used to enable integration with other ASD specifications:
- ‘Electronic_circuit_breaker’ (‘eltro’ in S1000D)
- ‘Electro_mechanic_circuit_breaker’ (‘elmec’ in S1000D)
- ‘Dummy_circuit_breaker’ (‘clip’ in S1000D)
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition_description</td>
<td>Descr</td>
<td>Textual description (explanation) of a specific instance of a condition used in one or many applicability statements.</td>
<td>Condition_by_identifier</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td>Condition_identifier</td>
<td>Id</td>
<td>Identification of a specific condition used in one or many applicability statements.</td>
<td>Condition_by_identifier</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td>Condition_name</td>
<td>Id</td>
<td>The name (title) by which a specific condition is known.</td>
<td>Condition_by_identifier</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td>Condition_type_class_value</td>
<td>Class</td>
<td>Valid classes to be used for a defined condition type.</td>
<td>Condition_type_class_value</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are examples which can be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Pre’ / ‘Post’ (Service_bulletin_condition_type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Ashore’ / ‘Afloat’ (Ashore_or_afloat_condition_type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Arctic’ / ‘Dessert’ (Operational_environment_condition_type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Docked’ / ‘Indoor’ / ‘Outdoor’ (Maintenance_environment_condition_type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Final valid classes are to be determined by the project.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condition_type_definition</td>
<td>Descr</td>
<td>Textual description of a specific condition type used in one or many applicability statements.</td>
<td>Condition_type</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Condition_type_name</td>
<td>Class</td>
<td>The name by which the condition type is known.</td>
<td>Condition_type</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are examples of classes which can be used to enable integration with other ASD Specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Service_bulletin_condition_type’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Ashore_or_afloat_condition_type’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Operational_environment_condition_type’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Maintenance_environment_condition_type’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condition_type_property_value</td>
<td>Prp</td>
<td>Valid property value for a defined condition type.</td>
<td>Condition_type_property_value</td>
<td>Applicability Statement</td>
</tr>
<tr>
<td>Contract_identifier</td>
<td>Id</td>
<td>Identification of a contract/subcontract established within the defined LSA project (LSA program).</td>
<td>Contract</td>
<td>Project</td>
</tr>
<tr>
<td>Contract_relationship_type</td>
<td>Class</td>
<td>Defines the type of relationship that is established between two contracts.</td>
<td>Contract_relationship</td>
<td>Project</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are recommended to be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Subcontract’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Related_contract’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data_module_code</td>
<td>Id</td>
<td>Identification of a specific data module created in accordance with the rules defined in S1000D.</td>
<td>S1000D_data_module</td>
<td>Document</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following Identifier_class must be used for Data_module_code, to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘S1000D_data_module_code’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Data_module_infoname</td>
<td>Id</td>
<td>The name by which a data module is known (S1000D element name = Infoname). The following Identifier_class must be used for Data_module_identifier, to enable integration with other ASD specifications: - ‘S1000D_data_module_infoname’</td>
<td>S1000D_data_module</td>
<td>Document</td>
</tr>
<tr>
<td>Data_module_issue_number</td>
<td>Id</td>
<td>Identification of a specific issue (revision) of a data module. The following Identifier_class must be used for Data_module_issue_number, to enable integration with other ASD specifications: - ‘S1000D_data_module_issue_number’</td>
<td>S1000D_data_module</td>
<td>Document</td>
</tr>
<tr>
<td>Detection_mean_description</td>
<td>Descr</td>
<td>Description of the mean that warns the user or maintainer that a failure has occurred.</td>
<td>Detection_mean</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Detection_mean_false_alarm_rate</td>
<td>Prp</td>
<td>The percent (expressed as a decimal value) of all alarms that indicate a malfunction that cannot be verified by maintenance personnel performing follow-on tests.</td>
<td>Detection_mean</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Detection_mean_type</td>
<td>Class</td>
<td>Defines if the test that generates the symptom is: 1) Power_on_built_in_test (P-BIT) – executed at start-up of the item, 2) Continuous_built_in_test (C-BIT) – periodically and automatically executed during the operation of the item, without any intervention from the operating crew, 3) Initiated_built_in_test (I-BIT) – executed upon order from the operator or from the maintenance team, 4) Ground_support_equipment.</td>
<td>Detection_mean</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Direct_maintenance_cost</td>
<td>Prp</td>
<td>For a given LSA candidate, the direct maintenance costs include shop maintenance manhours, shop test manhours, repair cost (incl. required material). Statistical values like MTBF and MTBUR are taken into consideration. The following Unit of Measures must be used to enable integration with other ASD specifications: - Subclass of 'Currency'</td>
<td>Direct_maintenance_cost</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Document_assignment_role</td>
<td>Class</td>
<td>Determination of the role of a document being assigned to something. The following classes are recommended to be used to enable integration with other ASD specifications: - 'Document_reference' - 'Drawing_reference' - 'Design_document_reference' - 'Directive' - 'Source' - 'Verification'</td>
<td>Document_assignment</td>
<td>Document</td>
</tr>
<tr>
<td>Document_identifier</td>
<td>Id</td>
<td>Identification assigned to a particular document.</td>
<td>External_document</td>
<td>Document</td>
</tr>
<tr>
<td>Document_issue_date</td>
<td>Date</td>
<td>Date of issue for a specific issue (revision) of a document.</td>
<td>External_document</td>
<td>Document</td>
</tr>
<tr>
<td>Document_issue_identifier</td>
<td>Id</td>
<td>Identification of a specific issue (revision) of a document.</td>
<td>External_document</td>
<td>Document</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Document_location</td>
<td>Descri</td>
<td>Description where to find a specific document (issue).</td>
<td>External_document</td>
<td>Document</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: May be a string that contains a hyperlink to the document (or specific issue of the document).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document_portion</td>
<td>Descri</td>
<td>Identifies a defined portion of a document which is of interest in a specific usage.</td>
<td>Document_assignment</td>
<td>Document</td>
</tr>
<tr>
<td>Document_title</td>
<td>Id</td>
<td>The title (name) by which a document is known.</td>
<td>External_document</td>
<td>Document</td>
</tr>
<tr>
<td>Document_type</td>
<td>Class</td>
<td>Classification of the document in order to indicate the type of document.</td>
<td>External_document</td>
<td>Document</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are recommended to be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Technical Report’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Standard’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Drawing’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Down_time</td>
<td>Prp</td>
<td>The acceptable (maximum) down time (MDT), where down time is the duration where an item is non-operational.</td>
<td>Down_time</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following Unit of Measures must be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Subclass of ‘Time_unit’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Estimated_number_of_lines_of_code</td>
<td>Prp</td>
<td>Estimated number of lines of executable code in a software that will</td>
<td>Software_element_revision</td>
<td>Breakdown Element Realization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>realize the software breakdown element.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following Unit of Measures must be used to enable integration with</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Lines_of_software_code'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated_software_complexity_code</td>
<td>Class</td>
<td>Rough classification of the estimated complexity of the software that</td>
<td>Software_element_revision</td>
<td>Breakdown Element Realization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>will realize the software breakdown element.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid classes are to be determined by the project.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated_software_modification_frequencies</td>
<td>Class</td>
<td>Estimated frequency of new releases (minor as well as major) for a</td>
<td>Software_element_revision</td>
<td>Breakdown Element Realization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>software that will realize the software breakdown element.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following Unit of Measures must be used to enable integration with</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Subclass of 'Time_unit'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated_software_size</td>
<td>Prp</td>
<td>Estimated size of a software that will realize the software breakdown</td>
<td>Software_element_revision</td>
<td>Breakdown Element Realization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>element</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following Unit of Measures is recommended to be used to enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Kilobyte'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Megabyte'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Gigabyte'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Terabyte'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Applicable to: All
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event_threshold_number_of_event_occurrences</td>
<td>int</td>
<td>The number of event occurrences required to initiate the related threshold or trigger.</td>
<td>Event_threshold</td>
<td>Task Usage (Part 1)</td>
</tr>
<tr>
<td>Failure_mode_description</td>
<td>Descr</td>
<td>A description of the manner by which a failure occurs.</td>
<td>Failure_mode</td>
<td>LSA-FMEA and Special</td>
</tr>
<tr>
<td>Failure_mode_detection_ability_description</td>
<td>Descr</td>
<td>Description of the ability to detect an identified failure mode. The description can include</td>
<td>Failure_mode</td>
<td>LSA-FMEA and Special</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a summary of the available detection means.</td>
<td></td>
<td>Events</td>
</tr>
<tr>
<td>Failure_mode_detection_ability_rating</td>
<td>Classifi</td>
<td>Defines the ability to detect an identified failure mode. Qualitative judgment made by the</td>
<td>Failure_mode</td>
<td>LSA-FMEA and Special</td>
</tr>
<tr>
<td></td>
<td>cation</td>
<td>analyst.</td>
<td></td>
<td>Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are recommended to be used:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'High likelihood'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Moderately_high_likelihood'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Medium likelihood'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Moderately_low_likelihood'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Low likelihood'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failure_mode_detection_rate</td>
<td>Prp</td>
<td>Defines the percentage of failure modes detected by the defined detection mean.</td>
<td>Detectability</td>
<td>LSA-FMEA and Special</td>
</tr>
<tr>
<td>Failure_mode_effect</td>
<td>Descr</td>
<td>A description of the consequences that a failure mode has on the local/next higher/end item</td>
<td>Failure_mode_effect</td>
<td>LSA-FMEA and Special</td>
</tr>
<tr>
<td></td>
<td></td>
<td>operation, function, or status.</td>
<td></td>
<td>Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Local effect' – The consequences of each postulated failure/damage mode affecting the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSA candidate must be described along with any second order effects that result. Potential</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>conditions where the failure/damage of one item results in a change of the conditional</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>failure or status.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Applicable to: All

DMC-S3000L-A-22-00-0000-00A-040A-A_001_00_EN-US.doc
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>probability or effect of a second item must be identified. It is possible for the &quot;local effect&quot; to be the failure/damage mode itself.</td>
<td></td>
<td></td>
<td>Class/interface</td>
<td>UoF</td>
</tr>
<tr>
<td>'Next higher effect' – The consequences of each failure/damage mode affecting the next higher indenture level must be described.</td>
<td></td>
<td></td>
<td>Class/interface</td>
<td>UoF</td>
</tr>
<tr>
<td>'End effect' – The effect of each failure/damage mode upon the essential functions(s) affecting system/equipment operating capability and mission completion capability must be determined. The end effect described may be the result of a double failure. For example, failure of a safety device may result in a catastrophic end effect only in the event that both the prime function goes beyond the limit for which the safety device is set, and the safety device fails.</td>
<td></td>
<td></td>
<td>Class/interface</td>
<td>UoF</td>
</tr>
<tr>
<td>Failure_mode_effect_level</td>
<td>Class</td>
<td>Failure effect level may be defined as either Next higher or End Item.</td>
<td>Failure_mode_effect</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>'Next higher effect' – These effects concentrate on the impact a failure/damage mode has on the operation and function of the items in the next higher indenture level above the LSA candidate under consideration.</td>
<td></td>
<td></td>
<td>Class/interface</td>
<td>UoF</td>
</tr>
<tr>
<td>'End effect' – End effects evaluate and define the total effect a failure/damage mode has on the operation, function, or status of the uppermost system.</td>
<td></td>
<td></td>
<td>Class/interface</td>
<td>UoF</td>
</tr>
<tr>
<td>Failure_mode_identifier</td>
<td>Id</td>
<td>Identification of a specific failure mode code, typically assigned by the contractor for an LSA candidate.</td>
<td>Failure_mode</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Failure_mode_isolation_rate</td>
<td>Prp</td>
<td>Defines the percentage of detected failures that are associated without ambiguity to the failure of an identified item or component.</td>
<td>Detectability</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Failure_mode_localization_abililty_description</td>
<td>Descri</td>
<td>Description of the ability to localize an identified failure mode. The description can include a summary of localization means available or required.                                                                                       Note: Failure mode localization_abililty_description is generally a complement to failure mode detection_abililty_description.</td>
<td>Failure_mode</td>
<td>LSA-FMEA and Special Events</td>
</tr>
</tbody>
</table>
| Failure_mode_localization_abililty_rating         | Classific | Defines the ability to localize an identified failure mode. Qualitative judgment made by the analyst. The following classes are recommended to be used:                                                                                 - ‘High_likelihood’  
- ‘Moderately_high_likelihood’  
- ‘Medium_likelihood’  
- ‘Moderately_low_likelihood’  
- ‘Low_likelihood’  | Failure_mode              | LSA-FMEA and Special Events |
<p>| Failure_rate                                      | Prp      | For a particular interval, the total number of failures within a population of an LSA candidate item divided by the total functional life of the population during the measurement interval. The definition holds for time, rounds, miles, cycles or other measures of life units.                                           Note: Ref GEIA-0007, failure_rate_type | Failure_rate              | LSA Candidate               |
| Failure_rate_correction_factor                    | Int      | Correction factor for the failure rate value. This can either be related to design or to the usage of the system under specific conditions eg environment which causes additional stress to the system (sand, extreme temperatures, salty environment).                                      | Failure_rate_correction   | LSA Candidate               |
| Failures_per_operating_hour                       | Prp      | Failure rate, expressed in failures per operating hour                                                                                                                                                                                                                                          | Failures_per_operating_hour | LSA Candidate               |</p>
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed_resource_marker</td>
<td>boolean</td>
<td>Indicates that no other resource than the specified is allowed to be used in the task/subtask.</td>
<td>Task_resource</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Functional_replaceability</td>
<td>Class</td>
<td>Classification whether a part is replaceable at its functional location. This classification is done from a technical standpoint (i.e. a vendor/supplier standpoint) and is independent of customer maintenance concepts. The following classes are recommended to be used to enable integration with other ASD specifications: - Replaceable - Not_replaceable - Not_applicable</td>
<td>Hardware_element_revision</td>
<td>Breakdown Element Revision</td>
</tr>
<tr>
<td>Hardware_element_type</td>
<td>Class</td>
<td>Can be used to indicate further specialization of a hardware element. The following classes are recommended to be used to enable integration with other ASD specifications: - Equipment - Access_point - Door, - Panel, - Electrical_panel</td>
<td>Hardware_element</td>
<td>Breakdown Element Realization</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Hardware_part_environmental_aspect_classes_in_use</td>
<td>Classification</td>
<td>Classification of environment aspects which needs to be considered during use of the hardware part. The following classes are recommended to be used to enable integration with other ASD specifications: - ‘Harmful_to_environment’ - ‘Acidification’ - ‘Dangerous_for_ozone_layer’ - ‘Greenhouse_effect’ - ‘Material_waste’ - ‘Energy_regaining_by_burning’ - ‘Material_recycling’</td>
<td>Hardware_part</td>
<td>Part</td>
</tr>
<tr>
<td>Hardware_part_environmental_aspect_classes_planned_disposal</td>
<td>Classification</td>
<td>Classification of environment aspects which needs to be considered during planned disposal of the hardware part. The following classes are recommended to be used to enable integration with other ASD specifications: - ‘Harmful_to_environment’ - ‘Acidification’ - ‘Dangerous_for_ozone_layer’ - ‘Greenhouse_effect’ - ‘Material_waste’ - ‘Energy_regaining_by_burning’ - ‘Material_recycling’</td>
<td>Hardware_part</td>
<td>Part</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>Hardware_part_logistics_category</td>
<td>Class</td>
<td>Defines the logistics role of a hardware part that is used within one or many tasks. The following classes are recommended to be used to enable integration with other ASD specifications:</td>
<td>Hardware_part</td>
<td>Part</td>
</tr>
</tbody>
</table>
|                                          |         | - ‘Support_equipment’  
|                                          |         | - ‘Common_support_equipment’  
|                                          |         | - ‘Special_support_equipment’  
|                                          |         | - ‘Spare’  
|                                          |         | - ‘Supply’  
|                                          |         | - ‘Consumable’  
|                                          |         | - ‘Expendable’  
|                                          |         | - ‘Material’  
|                                          |         | - ‘Facility’  
|                                          |         | - ‘Training_equipment’  
|                                          |         | Note 1: Equivalent to Item Category Code in GEIA-STD-0007.                                                                                                                                               |                 |      |
| Hardware_part_maintenance_start         | Class   | Defines when to initiate maintenance scheduling for a part. The following classes are recommended to be used to enable integration with other ASD specifications:                                             | Hardware_part   | Part |
|                                          |         | - ‘Maintenance_start_at_production’  
|                                          |         | - ‘Maintenance_start_at_delivery’  
|                                          |         | - ‘Maintenance_start_at_installation_in_assembly’  
<p>|                                          |         | - ‘Maintenance_start_at_installation_in_end_item’.                                                                                                                                                      |                 |      |</p>
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
</table>
| Hardware_part_material_hazardous_class           | Classifi cation | Identifies articles or substances which are capable of posing a significant risk to health, safety or property during transportation, handling or storage.  
Source: S2000M  
The Substance Identification Numbers listed in Chapter 2 of the United Nations Recommendations on the Transport of Dangerous Goods ST/SG/AC.10/1/Rev5, must be used to enable integration with other ASD specifications. | Hardware_part | Part |
| Hardware_part_operationalAuthorized_life         | Prp   | Authorized life indicates the maximum installed life for which an item may be operated. Upon reaching this authorized life, any further life of the item must be re-authorized.  
Source: S2000M | Hardware_part_operational_authorised_life | Part |
| Hardware_part_scrap_rate                         | Prp   | Indicates the estimated percentage of normally repairable units which, when removed from service, will be found to be beyond economic repair and therefore have to be scrapped.  
Note: The Unit of Measure must always be Percentage.  
Source: S2000M | Hardware_part | Part |
<p>| Hardware_part_waste_products_in_use_disposal_des cription | Descr | Describes how waste products for an individual part need to be managed when the part is disposed of according to the procedure during use or after being used. | Hardware_part | Part |</p>
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware_part_waste_products_planned_disposal_description</td>
<td>Descr</td>
<td>Describes how waste products need to be managed when the entire population of a part is disposed of according to the procedure for planned disposal.</td>
<td>Hardware_part</td>
<td>Part</td>
</tr>
<tr>
<td>Information_code</td>
<td>Class</td>
<td>Information code according to ASD S1000D. Information code is used for categorizing a task/subtask. S1000D information codes must be used to enable integration with other ASD specifications.</td>
<td>Task / Subtask_by_definition</td>
<td>Task</td>
</tr>
<tr>
<td>Installation_description</td>
<td>Descr</td>
<td>Description of the installation and location(s) for a breakdown element within a specific breakdown. Note: May be used to describe multiple installations, if Quantity_of_child_elements for a breakdown element is greater than one.</td>
<td>Breakdown_element_structure</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td>Key_performance_indicator_method</td>
<td>Descr</td>
<td>Describes the method by which the key performance indicator value has been derived.</td>
<td>Key_performance_indicator</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Key_performance_indicator_percentile</td>
<td>Prp</td>
<td>The percentage of all occurrences related to a specified key performance indicator that must be within the limit of the value defined for the key performance indicator. Example: A customer requirement is that 98% of all replacement tasks must be performed below a specified value of two hours (= maximum replacement time).</td>
<td>Key_performance_indicator</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Key_performance_indicator_status</td>
<td>Class</td>
<td>Status of acceptance of the recorded value for the key performance indicator. The following classes are recommended to be used: - ‘Preliminary’, - ‘Accepted’, - ‘Released’,</td>
<td>Key_performance_indicator</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Key_performance_indicator_type</td>
<td>Class</td>
<td>Defines the context in which the key performance indicator value is defined. The following classes are recommended to be used to enable integration with other ASD specifications: - ‘Specified’ (also known as allocated) - ‘Distributed’ - ‘Contracted’ - ‘Predicted’ - ‘Actual’.</td>
<td>Key_performance_indicator</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>LSA_candidate_indicator</td>
<td>Class</td>
<td>Defines the depth of analysis that will be performed on the candidate item.</td>
<td>LSA_candidate</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are recommended to be used:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Full candidate’, provide full scale of selected LSA information</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>applicable to the related item.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Partial candidate’, provide a partial scale of selected LSA information</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>applicable to the related item, eg only remove and install information</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>because of the need to gain access to other items, but no repair</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>information required, because the item is a discardable item.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Non’, no LSA required.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Open’, not yet decided.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSA_candidate_maintenance_concept</td>
<td>Descr</td>
<td>A statement of maintenance considerations, constraints and strategy for the</td>
<td>LSA_candidate</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>operational support that governs the maintenance levels and type of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>maintenance activities to be carried out for the LSA candidate under</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>analysis.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSA_candidate_maintenance_solution</td>
<td>Descr</td>
<td>A statement of maintenance activities and maintenance levels that has been</td>
<td>LSA_candidate</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>decided for the LSA candidate under analysis.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSA_candidate_rationale</td>
<td>Descr</td>
<td>Description of the reason for ‘Full’ / ’Partial’ / ‘Non’ LSA Candidate</td>
<td>LSA_candidate</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Indicator selection.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSA_failure_mode_description</td>
<td>Descr</td>
<td>Description of a generic failure mode which covers a set of individual</td>
<td>LSA_failure_mode</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>failure modes that has been grouped into the LSA failure mode.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSA_failure_mode_identifier</td>
<td>Id</td>
<td>Identification of an LSA failure mode.</td>
<td>LSA_failure_mode</td>
<td>LSA-FMEA and Special Events</td>
</tr>
</tbody>
</table>

Applicable to: All
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSA_failure_mode_distribution_occurrence_rating</td>
<td>Class</td>
<td>Identifies the rating for the occurrence of an individual LSA failure mode in comparison to the entire population of LSA failure modes identified for the item under analysis. The following classes are recommended to be used: - 'Moderately_high_likelihood' - 'Medium_likelihood' - 'Moderately_low_likelihood' - 'Low_likelihood' - 'Very_low_likelihood'</td>
<td>Distributed_LSA_failure_mode_with_rating LSA-FMEA and Special Events</td>
<td></td>
</tr>
<tr>
<td>LSA_failure_mode_distribution_ratio</td>
<td>Prp</td>
<td>Identifies the fraction of an individual LSA failure mode from the entire population of LSA failure modes identified for the item under analysis.</td>
<td>Distributed_LSA_failure_mode_with_rating LSA-FMEA and Special Events</td>
<td></td>
</tr>
<tr>
<td>Maintenance_free_operating_period</td>
<td>Prp</td>
<td>The acceptable (minimum) maintenance free operating period, where maintenance free operating period is the interval in which no maintenance actions occur.</td>
<td>Maintenance_free_operating_period LSA Candidate</td>
<td></td>
</tr>
<tr>
<td>Maintenance_level_type_capability_description</td>
<td>Descri</td>
<td>Description of a maintenance level type. May include textual descriptions of defined capabilities in terms of personnel, availability of special facilities, time limits and the environmental conditions to be assumed. These capabilities are the basis in determining the functions to be accomplished at each maintenance level.</td>
<td>Maintenance_level_type Product Usage</td>
<td></td>
</tr>
<tr>
<td>Maintenance_level_type_identifier</td>
<td>Id</td>
<td>Identification of a maintenance level authorized to perform a required maintenance function.</td>
<td>Maintenance_level_type Product Usage</td>
<td></td>
</tr>
<tr>
<td>Maintenance_level_type_name</td>
<td>Id</td>
<td>The name by which a maintenance level type is known.</td>
<td>Maintenance_level_type Product Usage</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Includes software maintenance levels.
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance_location_code</td>
<td>Class</td>
<td>Indicates where the maintenance task will be carried out in terms of a product (end item or major assembly). The following classes must be used to enable integration with other ASD specifications: “A” – Information related to items installed on the Product “B” – Information related to items installed on a major assembly removed from the Product “C” – Information related to items on the bench. In this context, it does not matter, for example, whether an item has been removed from the Product. “D” – Information related to all three locations A, B, and C. Source; S1000D Item Location Code</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Subtask_by_definition</td>
<td>Task</td>
</tr>
<tr>
<td>Maintenance_location_description</td>
<td>Descr</td>
<td>Description of a specific maintenance location. May include textual descriptions of actual capabilities in terms of personnel, availability of special facilities, time limits and the environmental conditions.</td>
<td>Maintenance_location</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Maintenance_location_identifier</td>
<td>Id</td>
<td>Identification of a specific maintenance location.</td>
<td>Maintenance_location</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Maintenance_location_name</td>
<td>Id</td>
<td>The name by which a specific maintenance location is known.</td>
<td>Maintenance_location</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Maintenance_man_hours_per_operating_hour</td>
<td>Prp</td>
<td>The (maximum) acceptable maintenance man hours per operating hour, where maintenance man hours per operating hour is the ratio of maintenance man hours expended to the operating interval (as defined by the measurement base) of the system/equipment.</td>
<td>Maintenance_man_hours_per_operating_hour</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Maintenance_significant_or_relevant_indicator</td>
<td>Class</td>
<td>Indicator, whether a breakdown element is classified as being a Maintenance Significant Item (MSI), Maintenance Relevant Item (MRI), or if the classification is Not Applicable (N/A). Note 1: An MSI is an item which was identified by any selection process coming from a Scheduled Maintenance Analysis Procedure like MSG-3 or ASD/AIA S4000M. For this item, a Scheduled Maintenance Analysis (SMA) will be performed. Note 2: An MRI is an item which can be repaired or replaced in case of failure or damage.</td>
<td>Breakdown_element_revision</td>
<td>Breakdown Structure</td>
</tr>
<tr>
<td>Material_characteristics_recording_date</td>
<td>Date</td>
<td>Date for latest changes of material characteristics data.</td>
<td>Material</td>
<td>Part</td>
</tr>
<tr>
<td>Material_description</td>
<td>Descr</td>
<td>Short description of material (substance) used in one or many parts.</td>
<td>Material</td>
<td>Part</td>
</tr>
<tr>
<td>Material_identifier</td>
<td>Id</td>
<td>Identification of the material (substance)</td>
<td>Material</td>
<td>Part</td>
</tr>
<tr>
<td>Material_name</td>
<td>Id</td>
<td>The name by which a material (substance) is known.</td>
<td>Material</td>
<td>Part</td>
</tr>
<tr>
<td>Material_risk_description</td>
<td>Descr</td>
<td>Description of risks associated with the material (substance).</td>
<td>Material</td>
<td>Part</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Material_risk_factor</td>
<td>Class</td>
<td>Classification of risk associated with the material (substance). Valid classes for material risk factors are to be determined by the project. Example: Example of risk factor is the judged risk factor of material in Shelf Life Analysis (SLA)</td>
<td>Material</td>
<td>Part</td>
</tr>
<tr>
<td>Material_substance_usage_category</td>
<td>Class</td>
<td>The main substances category. The following classes are recommended to be used: - ‘Forbden’ - ‘Authorized with limitation’ - ‘Authorized with no limitation’.</td>
<td>Material</td>
<td>Part</td>
</tr>
<tr>
<td>Material_usage_justification_description</td>
<td>Descr</td>
<td>Description of the most important property for the function of the part that the included material (substance) has.</td>
<td>Hardware_part_material_usage</td>
<td>Part</td>
</tr>
<tr>
<td>Mean_time_between_failure</td>
<td>Prp</td>
<td>The (minimum) mean time between failure (MTBF), where the MTBF is the total operational life of a population of an LSA candidate divided by the total number of failures within the population during a particular measurement interval. The definition holds for time, rounds, miles, events, or other measure of life units.</td>
<td>Mean_time_between_failure</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Mean_time_between_failure_correction_date</td>
<td>Date</td>
<td>Date for defining the correction factor.</td>
<td>Mean_time_between_failure_corr</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Mean_time_between_failure_correction_factor</td>
<td>int</td>
<td>Correction factor for the mean time between failure value. This can either be related to design (effects of eg no failure found, BIT cannot duplicate or detectability effects can influence the real applicable MTBF) or to the usage of the system under specific conditions eg environment which causes additional stress to the system (sand, extreme temperatures, salty environment).</td>
<td>Mean_time_between_failure_correction</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Mean_time_between_failure_correction_justification</td>
<td>Descr</td>
<td>Description of the reason for the introduction of the correction factor.</td>
<td>Mean_time_between_failure_correction</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Mean_time_between_unscheduled_removal</td>
<td>Prp</td>
<td>Minimum mean time between unscheduled removal, where mean time between unscheduled removal is the total number of operational units (eg, miles, rounds, hours) divided by the total number of items removed from that system during a stated period of time. This term is defined to exclude removals either being scheduled or performed to facilitate other maintenance and removals for product improvement.</td>
<td>Mean_time_between_unscheduled_removal</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Mean_time_to_repair</td>
<td>Prp</td>
<td>The (maximum) mean time to repair (MTTR), where MTTR is the total elapsed time for corrective maintenance divided by the total number of corrective maintenance actions during a given period of time.</td>
<td>Mean_time_to_repair</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Number_of_operating_locations</td>
<td>Prp</td>
<td>Number of locations which will receive and operate the contracted product variant.</td>
<td>Operating_location_type</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Operating_location_description</td>
<td>Descr</td>
<td>Description of an actual operating location.</td>
<td>Operating_location</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Operating_location_identifier</td>
<td>Id</td>
<td>Identification of an actual operating location.</td>
<td>Operating_location</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Operating_location_name</td>
<td>Id</td>
<td>The name by which an actual operating location is known.</td>
<td>Operating_location</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Operating_location_type_description</td>
<td>Descr</td>
<td>Description of an operating location type and its operating and environmental conditions to be assumed.</td>
<td>Operating_location_type</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Operating_location_type_identifier</td>
<td>Id</td>
<td>Identification of a type of operating location which will receive and operate the contracted product variant.</td>
<td>Operating_location_type</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Operating_location_type_name</td>
<td>Id</td>
<td>The name by which an operating location type is known.</td>
<td>Operating_location_type</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Operating_requirement_at_operating_location</td>
<td>Prp</td>
<td>Value of the (annual) operating requirement per operating location and contracted product variant.</td>
<td>Contracted_product_in_operating_location</td>
<td>Product Usage</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Note: Annual operating requirements may be measurement based, generally operating hours, but also other measurement bases can be possible.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating_requirement_at_operating_location_type</td>
<td>Prp</td>
<td>Value of the (annual) operating requirement per operating location type and contracted product.</td>
<td>Contracted_product_in_operating_location_type</td>
<td>Product Usage</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Note: Annual operating requirements may be measurement based, generally operating hours, but also other measurement bases can be possible.</em></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Applicable to: All
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization_assignment_role</td>
<td>Class</td>
<td>Determines the role of the organization being assigned to an object in the LSA database. The following classes are recommended to be used to enable integration with other ASD specifications: - ‘Authorizing_organization’ - ‘Design_responsible_organization’ - ‘Published_by_organization’</td>
<td>Organization_assignment</td>
<td>Organization_assignment</td>
</tr>
<tr>
<td>Organization_identifier</td>
<td>Id</td>
<td>Identification of an actual organization. Examples: NCAGE code, VAT number.</td>
<td>Organization</td>
<td>Project</td>
</tr>
<tr>
<td>Organization_name</td>
<td>Id</td>
<td>The name by which an actual organization is known.</td>
<td>Organization</td>
<td>Project</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>-----</td>
</tr>
</tbody>
</table>
| Part_demilitarization_class  | Classification| Part demilitarization identifies special measures to be taken when a part is being disposed of, eg:  
- render them useless for military purposes  
- destroy any indications of military purposes or performance characteristics  
*Source: S2000M*  
Usage of the following Part demilitarization classes are required to enable integration with other ASD specifications:  
- ‘Demilitarization not required’  
- ‘Trade Security Controls (TSC) required at disposal’  
- ‘Remove and/or demilitarize installed key point(s)’  
- ‘Demilitarize by mutilation’  
- ‘Demilitarization to be furnished by the MoD or national Demilitarization Program Office’  
- ‘Demilitarization instructions to be furnished by item/technical manager’  
- ‘Demilitarization required prior to transfer of item to national reutilization and disposition offices’  
- ‘Security Classified Item’ | Part            | Part |
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part_identifier</td>
<td>Id</td>
<td>Any identifier of an item of production (including software). Includes manufacturer’s part, drawing, model, type, or source controlling numbers. Usage of the following Part_identifier classes are recommended to enable integration with other ASD specifications: - ‘ASD_system_or_hardware_identification_code’, - ‘Nato_stock_number’, - ‘OEM_part_identification_code’, - ‘Supplier_part_identification_code’;</td>
<td>Part</td>
<td>Part</td>
</tr>
<tr>
<td>Part_maturity</td>
<td>Classifi cation</td>
<td>Classification defining the maturity of the part in order to determine the certainty by which the parts characteristics can be valued. The following classes are recommended to be used to enable integration with other ASD specifications: - ‘New_developed’ - ‘Major_modification_of_existing_item’ - ‘Moderate_modification_of_existing_item’ - ‘COTS_item’ - ‘GSE/GFE_items’</td>
<td>Part</td>
<td>Part</td>
</tr>
<tr>
<td>Part_name</td>
<td>Id</td>
<td>The name by which a part is known. The following Part_name Identifier_class is recommended to be used to enable integration with other ASD specifications: - ‘Technical name’ (techname in S1000D).</td>
<td>Part</td>
<td>Part</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------</td>
</tr>
<tr>
<td>Part_obsolescence_risk_assessment</td>
<td>descr</td>
<td>Describes risk associated with loss of the part in the supply chain due to factors such as termination of manufacturing, replacement with a different model, etc.</td>
<td>Part</td>
<td>Part</td>
</tr>
<tr>
<td>Part_special_handling</td>
<td>descr</td>
<td>Describes any requirements for special handling of the part.</td>
<td>Part</td>
<td>Part</td>
</tr>
<tr>
<td>Parts_list_entry_identifier</td>
<td>id</td>
<td>Any identifier of an entry in a parts list (Bill-Of-Material). Usage of the following Parts_list_entry_identifier classes are required to enable integration with other ASD specifications: - ‘Catalogue_sequence_number’</td>
<td>Parts_list_entry</td>
<td>Part</td>
</tr>
<tr>
<td>Parts_list_entry_substitute_identifier</td>
<td>id</td>
<td>The ‘Parts_list_entry_substitute_identifier’ together with the ‘Parts_list_entry_identifier’ can provide a unique key for each record in a Parts_list_entry (Bill-of-Material). Usage of the following Parts_list_entry_substitute_identifier class is recommended to enable integration with other ASD specifications: - ‘Item_sequence_number’ (requires the Parts_list_identifier to be classified as Catalogue_sequence_number)</td>
<td>Substitute_part_relationship</td>
<td>Part</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>-------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>------</td>
</tr>
</tbody>
</table>
| Physical_replaceability | Class | Classification of whether a part is interchangeable in a specific assembly. This classification is done from a technical standpoint (i.e. a vendor/supplier standpoint) and is independent of customer maintenance concepts.  

The following classes are recommended to be used to enable integration with other ASD specifications:  
- ‘Interchangeable’ (Interchangeable items must be capable of being readily installed, removed, or replaced without alteration misalignment or damage to items being installed or to adjoining items or structure)  
- ‘Replaceable’ (Replaceable items requires alterations of the items in addition to the normal application and methods of attachment. Such alterations may include drilling, reaming, cutting, filing, trimming, bending, shaping, etc)  
- ‘Not_applicable’  

*Source: EIA-836 Configuration Management* | Parts_list_entry | Part |
|-------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|------|
| Product_identifier | Id   | Identification of a product which is the target for the Logistics Support Analysis Project (Program).  

*Note 1: Equivalent to End Item Acronym Code in GEIA-STD-0007.*  

*Note 2: May be used to capture ASD S1000D Model Identification Code, where the Model Identification Code is used for more than one Product Variant.* | Product | Project |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Product_name</td>
<td>Id</td>
<td>The name by which the Product is known.</td>
<td>Product</td>
<td>Project</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Product_service_life</td>
<td>Prp</td>
<td>The number of years the LSA candidate is expected to be in service. Note: Other measures than years can be used.</td>
<td>Product_service_life</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Product_usage_phase</td>
<td>Class</td>
<td>Defined phases during the products in-service phase. Examples of product usage phases are: - 'Operation' - 'Take off' - 'Flight' - 'Landing' - 'Maintenance' - 'Storage' - 'Transportation' Valid classes are to be determined by the project.</td>
<td>Product_usage_phase</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Product_variant_identifier</td>
<td>Id</td>
<td>Identification of a specific variant (model) of the contracted product being the target for the Logistics Support Analysis Project (Program). Note 1: Equivalent to System/End Item Usable On Code in GEIA-STD-0007. The following Product_variant_identifier is recommended to be used to enable integration with other ASD specifications: - 'Model_identification_code'.</td>
<td>Product_variant</td>
<td>Project</td>
</tr>
<tr>
<td>Product_variant_name</td>
<td>Id</td>
<td>The name by which the product variant (model) is known.</td>
<td>Product_variant</td>
<td>Project</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Product_variant_realization_identifier</td>
<td>Id</td>
<td>Identification of a product variant realization (i.e. an allowed physical configuration). A product variant realization identifies the approved parts and software which are allowed to realize the individual breakdown elements defined for a product breakdown.</td>
<td>Product_variant_realization</td>
<td>Product Variant Applicability</td>
</tr>
<tr>
<td>Product_variant_realization_revision_creation_date</td>
<td>Date</td>
<td>Date for creation of a revision of a product variant realization (i.e. an allowed physical configuration).</td>
<td>Product_variant_realization</td>
<td>Product Variant Applicability</td>
</tr>
<tr>
<td>Product_variant_realization_revision_identifier</td>
<td>Id</td>
<td>Identification of a specific revision of a product variant realization. <strong>Note:</strong> Revision identifier is used to identify design iterations and not variants.</td>
<td>Product_variant_realization</td>
<td>Product Variant Applicability</td>
</tr>
<tr>
<td>Project_identifier</td>
<td>Id</td>
<td>Identification of the project (program) within which the Logistics Support Analysis (LSA) is being performed.</td>
<td>Project</td>
<td>Project</td>
</tr>
<tr>
<td>Project_name</td>
<td>Id</td>
<td>The name by which the project (LSA program) is known.</td>
<td>Project</td>
<td>Project</td>
</tr>
<tr>
<td>Publication_module_code</td>
<td>Id</td>
<td>Identification of a specific publication module created in accordance with the rules defined in S1000D.</td>
<td>S1000D_publication_module</td>
<td>Document</td>
</tr>
<tr>
<td>Publication_module_issue_number</td>
<td>Id</td>
<td>Identification of a specific issue (revision) of a publication module.</td>
<td>S1000D_publication_module</td>
<td>Document</td>
</tr>
<tr>
<td>Publication_module_title</td>
<td>Id</td>
<td>The name by which a publication module is known.</td>
<td>S1000D_publication_module</td>
<td>Document</td>
</tr>
<tr>
<td>Quantity_of_child_elements</td>
<td>Prp</td>
<td>The quantity of a child element in the context of its parent element.</td>
<td>Breakdown_element_structure / Part_list_entry</td>
<td>Breakdown Structure / Part</td>
</tr>
<tr>
<td>Quantity_of_contracted_product_variant</td>
<td>Prp</td>
<td>Quantity of a product variant included in a contract.</td>
<td>Contracted_product_variant</td>
<td>Project</td>
</tr>
</tbody>
</table>

Applicable to: All
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity_of_material_included_in_hardware_part</td>
<td>Prp</td>
<td>Quantity of a material included in a hardware part.</td>
<td>Hardware_part_material_usage</td>
<td>Part</td>
</tr>
<tr>
<td>Quantity_of_product_variant_in_operating_location</td>
<td>Prp</td>
<td>The number of serialized items, of a specific product variant, that is to be operated at an actual operating location.</td>
<td>Contracted_product_variant_in_operating_location</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Quantity_of_product_variant_in_operating_location_type</td>
<td>Prp</td>
<td>The number of serialized items, of a specific product variant, that is to be operated at a specific operating location type.</td>
<td>Contracted_product_variant_in_operating_location_type</td>
<td>Product Usage</td>
</tr>
<tr>
<td>Remark_text</td>
<td>Descr</td>
<td>Text that provides the user with information that is helpful but does not belong to the immediate subject.</td>
<td>Remark</td>
<td>Remark</td>
</tr>
<tr>
<td>Remark_type</td>
<td>Class</td>
<td>Additional characterization of the remark, defining its purpose. The following class can be used to enable integration with other ASD specifications: <code>Maintenance_allocation_remark</code> (S1000D Maintenance Allocation)</td>
<td>Remark</td>
<td>Remark</td>
</tr>
</tbody>
</table>

*Source: OASIS PLCS Standard Reference Data Library*
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repairability</td>
<td>Class</td>
<td>Classification whether a part is repairable from a technical standpoint (i.e. a vendor/supplier standpoint). This classification is independent from customer maintenance concepts. The following classes must be used to enable integration with other ASD specifications: - ‘Reapairable’ - ‘Partially_repairable’ - ‘Discardable’ - ‘Not_applicable’.</td>
<td>Hardware_part / Hardware_element_revision</td>
<td>Part / Breakdown Element Realization</td>
</tr>
<tr>
<td>Repairability_strategy</td>
<td>Class</td>
<td>Classification of the repairability strategy chosen for a specific customer and maintenance concept. Repairability strategy can include information on which maintenance level the repair is to be performed. Valid classes for Repairability_strategy are to be determined by the project.</td>
<td>Hardware_part / Hardware_element_revision</td>
<td>Part / Breakdown Element Realization</td>
</tr>
<tr>
<td>Replaceability_strategy</td>
<td>Class</td>
<td>Defines the replaceability strategy chosen for a specific customer and maintenance concept. Replaceability strategy can include information about the maintenance level, at which the replacement task is to be performed. Valid classes for Replaceability_strategy are to be determined by the project. Examples on classes that can be used are: - ‘Line_replaceable’, - ‘Shop_replaceable’.</td>
<td>Parts_list_entry / Hardware_element_revision</td>
<td>Part / Breakdown Element Revision</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Replacement_time</td>
<td>Prp</td>
<td>The duration of the replacement of a (eg faulty) component within any technical system by another (eg new) component.</td>
<td>Replacement_time</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Resource_specification_description</td>
<td>Descri</td>
<td>Textual specification of a resource needed to perform a task.</td>
<td>Resource_specification</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Resource_specification_identifier</td>
<td>Id</td>
<td>Identification of a resource specification.</td>
<td>Resource_specification</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Resource_specification_name</td>
<td>Id</td>
<td>Name by which the resource being specified is known.</td>
<td>Resource_specification</td>
<td>Task Resources</td>
</tr>
<tr>
<td>S1000D_task_type</td>
<td>Class</td>
<td>Defines how the task corresponds with the schemas defined in S1000D.</td>
<td>Task</td>
<td>Task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following S1000D_task_types must be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Procedure',</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Package'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling_method_description</td>
<td>Descri</td>
<td>Description of sampling method used.</td>
<td>Sampling</td>
<td>Task Usage (Part 1)</td>
</tr>
<tr>
<td>Sampling_method_ratio</td>
<td>double</td>
<td>The number of samples over the total population expressed as a ratio, eg 0.1 (equals 10%)</td>
<td>Sampling_by_ratio</td>
<td>Task Usage (Part 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The value must be represent as decimal, even though it should have been a fraction.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling_method_value</td>
<td>Prp</td>
<td>The number of samples over the total population expressed as a value, eg ten aircrafts.</td>
<td>Sampling_by_value</td>
<td>Task Usage (Part 1)</td>
</tr>
<tr>
<td>Scheduled_maintenance_interval</td>
<td>Prp</td>
<td>The (minimum) number of operational units (eg rounds, miles, hours) between scheduled maintenance.</td>
<td>Scheduled_maintenance_interval</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Security_class</td>
<td>Class</td>
<td>Security class associated with a specific, breakdown element, part, task, subtask or task requirement. Valid classes for Security_class are to be determined by the project.</td>
<td>Security_class</td>
<td>Security Classification</td>
</tr>
<tr>
<td>Serial_number_lower_bound</td>
<td>int</td>
<td>The lower bound of a specified range of serialized items, as known by the customer.</td>
<td>Block_of_serialized_items</td>
<td>Project</td>
</tr>
<tr>
<td>Serial_number_upper_bound</td>
<td>int</td>
<td>The upper bound of a specified range of serialized items, as known by the customer.</td>
<td>Block_of_serialized_items</td>
<td>Project</td>
</tr>
<tr>
<td>Shop_processing_time</td>
<td>Prp</td>
<td>The duration from the start of the repair activities in the repair shop until the closing of the repair procedure without considering any shipping and delay times.</td>
<td>Shop_processing_time</td>
<td>LSA Candidate</td>
</tr>
<tr>
<td>Skill_code</td>
<td>Class</td>
<td>Identification of a specific combination of trade and skill level. Valid classes for Skill_code are to be determined by the project.</td>
<td>Skill</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Skill_level_name</td>
<td>Class</td>
<td>A defined competency level. The following classes are recommended to be used to enable integration with other ASD specifications: - ‘Advanced’ - ‘Intermediate’ - ‘Basic’.</td>
<td>Skill_level</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Software_element_type</td>
<td>Class</td>
<td>Can be used to indicate further specialization of a software element. The following classes are recommended to be used to enable integration with other ASD specifications: - ‘Loadable’ - ‘Embedded’.</td>
<td>Software_element</td>
<td>Breakdown Element Realization</td>
</tr>
<tr>
<td>Software_release</td>
<td>Id</td>
<td>Identification of a release (revision) of a manufactured software.</td>
<td>Software_part</td>
<td>Part</td>
</tr>
<tr>
<td>Software_technology</td>
<td>Desrc</td>
<td>Description of technology used for the development or execution of a manufactured software. Used for harmonization purposes, i.e. harmonizing the software technologies used within a product.</td>
<td>Software_part</td>
<td>Part</td>
</tr>
<tr>
<td>Software_type</td>
<td>Class</td>
<td>Classification defining if the software and/or data can be loaded/unloaded to/from the related hardware element (part). The following classes are recommended to be used to enable integration with other ASD specifications: - ‘Loadable’ - ‘Embedded’ - ‘Distributed’.</td>
<td>Software_part</td>
<td>Part</td>
</tr>
<tr>
<td>Special_event_description</td>
<td>Desrc</td>
<td>Description of a special event that can cause a related failure mode (damage).</td>
<td>Special_event</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Special_event_effect_probability_ratio</td>
<td>double</td>
<td>Identifies the fraction of a specific failure mode (damage) to occur out of the total number of failure modes (damages) that can occur as a consequence of the special event affecting a specific LSA candidate.</td>
<td>Special_event_effect_probability</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------------</td>
<td>---------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>Special_event_group</td>
<td>Class</td>
<td>Classification used for the grouping of special events. The following classes are recommended to be used to enable integration with other ASD specifications: - ‘External cause’ - ‘Natural phenomenon’ - ‘Meteorological’ - ‘Animal’ - ‘Human impact’ - ‘Combat’ - ‘Material maneuver’ - ‘Internal cause’ - ‘Internal dysfunction’ - ‘Extensive heat’ - ‘Extensive vibration’</td>
<td>Special_event</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Special_event_occurrence_rate</td>
<td>Prp</td>
<td>Quantification of how often a specific special event will occur.</td>
<td>Quantified_special_event_occurrence_definition</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Special_event_occurrence_rating</td>
<td>Class</td>
<td>Qualification of how often a specific special event will occur. The following classes is proposed to be used to enable integration with other ASD specifications: - ‘Extremely_unlikely’ - ‘Remote_likelihood’ - ‘Occasional’ - ‘Reasonably_probable’ - ‘Frequent’</td>
<td>Rated_special_event_occurrence_definition</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Special_event_title</td>
<td>Class</td>
<td>Title (name) by which a special event is known.</td>
<td>Special_event</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td>Special_training_requirement_description</td>
<td>Descr</td>
<td>Description of additional training required for the given competence (trade or skill).</td>
<td>Special_training_requirement</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Structural_indicator</td>
<td>Class</td>
<td>Indicator, whether a hardware breakdown element is classified as being a Structural Significant Item (SI), Structural Item (SI), Structural Detail (SD), or if the classification is Not Applicable (N/A). Note 1: A Structure Significant Item (SSI) is an item which was identified by any selection process coming from a Scheduled Maintenance Analysis Procedure like MSG-3 or ASD/AIA S4000M. For this item, a Scheduled Maintenance Analysis (SMA) will be performed. Note 2: A Structural Item (SI) is part of the systems bodywork. Note 3: A Structural Detail (SD), also named Significant Detail in the ASD S4000M, is a limited area of an SSI or a local spot being also part of the whole SSI. In contrast to an SSI, the SD isn’t identified by an own identifier (eg an own part number). Therefore these SD’s must be defined and documented on the respective SSI drawing. For identification within any breakdown structure, an additional artificial breakdown element identifier can be addressed.</td>
<td>Hardware_breakdown_element</td>
<td>Breakdown Element Realization</td>
</tr>
<tr>
<td>Subtask_acceptance_parameter_description</td>
<td>Descr</td>
<td>Description of criteria that determines whether a subtask is completed.</td>
<td>Subtask_acceptance_parameter</td>
<td>Task</td>
</tr>
<tr>
<td>Subtask_acceptance_parameter_value</td>
<td>Prp</td>
<td>Value (criteria) that must be fulfilled in order to complete the subtask.</td>
<td>Subtask_acceptance_parameter</td>
<td>Task</td>
</tr>
<tr>
<td>Subtask_description</td>
<td>Descr</td>
<td>Detailed description of the subtask procedure.</td>
<td>Subtask_by_definition</td>
<td>Task</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Subtask_duration</td>
<td>Prp</td>
<td>The average time expended, regardless of the number of personnel working simultaneously, required for the performance of the subtask. This does not include time spent awaiting spares, support equipment, facilities or personnel (logistics delay time).</td>
<td>Subtask_by_definition</td>
<td>Task</td>
</tr>
<tr>
<td>Subtask_identifier</td>
<td>Id</td>
<td>Identification of a subtask. [Note: A subtask is identified within the context of a task.]</td>
<td>Subtask</td>
<td>Task</td>
</tr>
<tr>
<td>Subtask_name</td>
<td>Id</td>
<td>The name (title) by which a subtask is known.</td>
<td>Subtask_by_definition</td>
<td>Task</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Subtask_objective_state</td>
<td>Class</td>
<td>Defines the new state that will exist due to the accomplishment of the subtask.</td>
<td>Subtask_objective_state</td>
<td>Task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes must be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Task_checked’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Jacked’ / ‘Unjacked’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Safety_device_established’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Electrical_power_established’ / ‘Electrical_power_from_engine_established’ / ‘Electrical_power_from_apu_established’ / ‘External_electrical_power_established’ / ‘Internal_electrical_power_established’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Hydraulic_power_established’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Air_supply_established’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Fueled’ / ‘Defueled’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Water_supply_established’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Controls_status_established’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Open’ / ‘Close’ / ‘Verify_open’ / ‘Verify_close’ (for Subtask_circuit_breaker_state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Subtask_role</td>
<td>Class</td>
<td>Defines if the subtask is required for preparation, core, or close-up purposes.</td>
<td>Subtask</td>
<td>Task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes must be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Startup’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Core’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Core_no_required_conditions’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Closeup’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtask_timeline_event</td>
<td>Class</td>
<td>Defining the starting point for subtask under consideration, in relation to the start or end point of the subtask playing the role of its predecessor.</td>
<td>Subtask_timeline</td>
<td>Task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes are recommended to be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Start’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘End’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtask_timeline_lag</td>
<td>Prp</td>
<td>Time between the related subtask timeline event (start / end) and the start point for the subtask under consideration.</td>
<td>Subtask_timeline</td>
<td>Task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Time lining just using references to the start point for the entire task, can use the lag attribute to indicate the time relative to the starting point of the first subtask and set Subtask_timeline_event = ‘Start’.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>-------</td>
</tr>
</tbody>
</table>
| Task_criticality_code                | Class    | Indicates whether or not the task is critical. A task is critical if failure to accomplish it would result in adverse effects on system reliability, efficiency, effectiveness, safety, or cost. A task will also be designated as critical whenever system design characteristics approach human limitations, and thereby, significantly increase the likelihood of degraded, delayed, or otherwise impaired mission performance. Valid classes for Task_criticality_code are to be determined by the project.  
   *Note: GEIA-STD-0007 uses the values ‘Yes’ / ’No’* | Task           | Task |
| Task_distribution_ratio              | double   | The probability of a specific task to be performed, when there are multiple Tasks that can resolve the same task requirement                                                                                                                                                                                                                           | Task_distribution | Task |
| Task_duration                        | Prp      | The average time expended, regardless of the number of personnel working simultaneously, required for the performance of a task, scheduled or unscheduled. This does not include time spent awaiting spares, support equipment, facilities or personnel (logistics delay time).  
   *Note: May be calculated from the subtask durations.*                                                                                                                               | Task            | Task |
<p>| Task_facility_resource_quantity      | Prp      | Quantity of a specific facility resource required by a subtask, or aggregated per task.                                                                                                                                                                                                                                                       | Task_facility_resource | Task Resources |
| Task_frequency                       | Prp      | The frequency of performance or occurrence of the task, expressed as the number of annual occurrences.                                                                                                                                                                                                                                | Task_frequency    | Task Usage (Part 1) |
| Task_frequency_calculation_method    | Descr    | Description of the method used for calculating the task frequency.                                                                                                                                                                                                                                                                        | Task_frequency    | Task Usage (Part 1) |</p>
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
</table>
| Task_identifier                                 | Id   | Identification of a task.  
              *Note: If the Task_identifier is being used in conjunction with S1000D it must follow the rules of XML identifier datatypes (start with an alpha character)* | Task            | Task         |
| Task_limit_description                         | Descr| Textual description of the limit (threshold and triggers) for performing a defined task. | Task_limit      | Task Usage (Part 1) |
| Task_limit_harmonization_indicator             | boolean | Marker for task limit, which is the result from a task limit harmonization and packaging procedure. Such tasks are in general inspection or/and overhaul packages, documented against non physical breakdown elements. | Task_limit      | Task Usage (Part 1) |
| Task_material_resource_category                | Class | Classification of the needed material resource as being eg spare part, supply, or support equipment in the context of the task/subtask.  
              The following classes must be used to enable integration with other ASD specifications:  
              - `Support_equipment`  
              - `Safety_related_support_equipment`  
              - `Spare`  
              - `Supply`. | Task_material_resource | Task Resources          |
| Task_material_resource_quantity                 | Prp  | Quantity of the material resource needed by a subtask, or aggregated per task. | Task_material_resource | Task Resources |

Applicable to: All
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task_name</td>
<td>Id</td>
<td>The name (title) by which a task is known.</td>
<td>Task</td>
<td>Task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following rule must be followed to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Task_name must use the information code name for the Information_code that is associated with the task</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Additional qualifiers can be added to establish a unique Task_name if multiple tasks with the same Information_code is associated with one and the same breakdown element or part.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task_number_of_personnel_resource</td>
<td>Prp</td>
<td>Number of persons with the same role and skill code needed within one task/subtask.</td>
<td>Task_personnel_resource</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Task_operability_code</td>
<td>Class</td>
<td>A code used to indicate the operational status and mission readiness of the end item during the maintenance task.</td>
<td>Task</td>
<td>Task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes must be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>“A” – System Inoperable during Equipment Maintenance,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>“B” – System Operable during Equipment Maintenance,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>“C” – Full Mission Capable,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>“D” – Partial Mission Capable,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>“E” – Not Mission Capable,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>“F” – Turnaround</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Note: Ref GEIA-0007, except code = G.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task_personnel_resource_labour_time</td>
<td>Prp</td>
<td>Time expended within a task/subtask per needed human resource.</td>
<td>Task_personnel_resource</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Task_personnel_resource_role</td>
<td>Class</td>
<td>Role of a specific instance of a personnel resource needed within a task/subtask.</td>
<td>Task_personnel_resource</td>
<td>Task Resources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes must be used to enable integration with other ASD specifications: &quot;Man A&quot; / &quot;Man B&quot; etc</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Performer'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Supervisor'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'Quality_assurance'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task_requirement_authority_source_type</td>
<td>Class</td>
<td>Indicates the source of an authority driven task requirement.</td>
<td>Authority_driven_task_requirement</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes must be used to enable integration with other ASD specifications: &quot;MSG3&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'CMR'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'AD'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task_requirement_change_description</td>
<td>Descr</td>
<td>Description of the changes introduced between two revisions of a task requirement.</td>
<td>Task_requirement_change</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Task_requirement_date</td>
<td>Date</td>
<td>The date when the task requirement was created.</td>
<td>Task_requirement</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Task_requirement_decision</td>
<td>Class</td>
<td>The status for the task requirement. Valid classes are to be determined by the project. Proposed classes to be used includes: 'Accepted', 'Rejected', 'Deferred', 'Realized'.</td>
<td>Task_requirement</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Task_requirement_description</td>
<td>Descri</td>
<td>Description of the maintenance requirements that have been determined as the outcome of eg the Reliability Centered Maintenance (RCM) analysis, LSA-FMEA, In-service feedback etc.</td>
<td>Task_requirement</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Task_requirement_identifier</td>
<td>Id</td>
<td>Identification of a task requirement.</td>
<td>Task_requirement</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Task_requirement_revision_identifier</td>
<td>Id</td>
<td>Identification of a revision of a task requirement. <strong>Note</strong>: Revisions is to be seen as design iterations and not as variants.</td>
<td>Task_requirement</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Task_requirement_special_resource</td>
<td>Descri</td>
<td>Textual description of any special (peculiar) resource needed for the performance of the required task.</td>
<td>Task_requirement</td>
<td>LSA Candidate Task Requirement</td>
</tr>
<tr>
<td>Task_resource_duration</td>
<td>Prp</td>
<td>The average time that a resource is needed for the performance of a task, scheduled or unscheduled.</td>
<td>Task_resource</td>
<td>Task Resources</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>--------------------------</td>
</tr>
</tbody>
</table>
| Task_resource_relationship_category            | Class    | Determines the relationship between two resources needed within the same subtask.  
Example:  
- Person A uses Support Equipment B  
Valid classes for Task_resource_relationship_category are to be determined by the project.                                                                                     | Task_resource_relationship       | Task Resources           |
| Task_revision_change_description               | Desctag  | Description of the changes introduced between two revisions of a task.                                                                                                                                                                                                                                                                     | Task_change                      | Task                     |
| Task_revision_identifier                       | Id       | Identification of a revision of a task.  
Note: Revisions is to be seen as design iterations and not as variants.                                                                                                                                                                                                                                                                  | Task                             | Task                     |
| Task_revision_status                           | Class    | Code for indicating the progress on the definition of a task (revision).  
Valid classes for Task_revision_status are to be determined by the project.                                                                                                                                                                                                       | Task                             | Task                     |
| Task_total_labour_time                         | Prp      | Total time expended within a task. Includes the labor time for all required personnel resources.                                                                                                                                                                                                                                      | Task                             | Task                     |
| Technology_behaviour_knowledge_rating          | Classifciation | Knowledge of the behavior of the technology used, during normal use or caused by special event  
Valid classes for Technology_behaviour_knowledge_rating are to be determined by the project.                                                                                                                | LSA_candidate_technology_behavior_rating | LSA-FMEA and Special Events |
<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class/Interface</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology_sensitivity_rating</td>
<td>Classification</td>
<td>Sensitivity regarding possible damage sources during normal use or caused by special event.</td>
<td>LSA_candidate_technology_behavior_rating</td>
<td>LSA-FMEA and Special Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid classes for Technology_sensitivity_rating are to be determined by the project.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threshold_value</td>
<td>Prp</td>
<td>A value that defines the limit (interval) for when a task is to be performed.</td>
<td>Parameter_threshold</td>
<td>Task Usage (Part 1)</td>
</tr>
<tr>
<td>Trade_name</td>
<td>Class</td>
<td>Type or classification of occupation.</td>
<td>Trade</td>
<td>Task Resources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid classes for Trade_name are to be determined by the project.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training_method</td>
<td>Class</td>
<td>Defines how training is to be performed.</td>
<td>Special_training_requirement</td>
<td>Task Resources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following classes can be used to enable integration with other ASD specifications:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Classroom’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Correspondence’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Maneuver area’</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ‘Simulator’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Warning_caution_or_note_description</td>
<td>Descr</td>
<td>Advices concerning safety, legal and health aspects.</td>
<td>Warning_caution_or_note</td>
<td>Task</td>
</tr>
<tr>
<td>Warning_caution_or_note_identifier</td>
<td>Id</td>
<td>Identification of the advices concerning safety, legal and health aspects.</td>
<td>Warning_caution_or_note</td>
<td>Task</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class/Interface</td>
<td>UoF</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>-----------</td>
</tr>
</tbody>
</table>
| Warning_caution_or_note_type | Class  | Classifies the provided safety, legal and health aspects as a warning, caution or note.  
The following classes must be used to enable integration with other ASD specifications:  
- ‘Warning’  
- ‘Caution’  
- ‘Note’. | Warning_caution_or_note      | Task                  |
| Zone_element_description     | Descr  | Description of the zone element under consideration.                       | Zone_element_revision | Breakdown Zone Element |
| Zone_element_type            | Class  | Can be used to indicate further specialization of a zone element.           | Zone_element          | Breakdown Zone Element |
Table 3 Attributes of the S3000L data types

<table>
<thead>
<tr>
<th>Data element</th>
<th>Type</th>
<th>Definition</th>
<th>Class</th>
<th>UoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class_name</td>
<td>char</td>
<td>The word or code by which the Class is known.</td>
<td>Class Data Types</td>
<td></td>
</tr>
<tr>
<td>Class_rdl</td>
<td>char</td>
<td>An identifier for the external class library, where a definition of the class is to be found.</td>
<td>Class Data Types</td>
<td></td>
</tr>
<tr>
<td>Class_rdl</td>
<td>char</td>
<td>Note: If appropriate this must be the URL/URN of the external class library.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification_date</td>
<td>Date</td>
<td>The date when a class was assigned to the subject under consideration.</td>
<td>Classification Data Types</td>
<td></td>
</tr>
<tr>
<td>Day_component</td>
<td>int</td>
<td>The year element of the Date expressed as a value between 1 and 31.</td>
<td>Date Data Types</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>char</td>
<td>Text that provides further information about the subject under consideration.</td>
<td>Descri Data Types</td>
<td></td>
</tr>
<tr>
<td>Description_provided_by_organization</td>
<td>Id</td>
<td>Defines the organization that provided the description.</td>
<td>Descri Data Types</td>
<td></td>
</tr>
<tr>
<td>Description_provided_by_organization</td>
<td>Id</td>
<td>Source: ISO 10303:239 Product Life Cycle Support.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description_provided_date</td>
<td>Date</td>
<td>The date when the description was provided.</td>
<td>Descri Data Types</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>char</td>
<td>The text that conveys the assigned identifier.</td>
<td>Id Data Types</td>
<td></td>
</tr>
<tr>
<td>Identifier_class</td>
<td>Id</td>
<td>Defines the type of identifier being defined, eg OEM part number, NCAGE-code etc.</td>
<td>Id Data Types</td>
<td></td>
</tr>
<tr>
<td>Identifier_class</td>
<td>Id</td>
<td>Source: ISO 10303:239 Product Life Cycle Support.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identifier_set_by_organization</td>
<td>Id</td>
<td>Defines the organization that “owns” the assigned identifier.</td>
<td>Id Data Types</td>
<td></td>
</tr>
<tr>
<td>Limit_qualifier</td>
<td>Class</td>
<td>The kind of limit. Valid values are ‘maximum’ or ‘minimum’</td>
<td>Value_with_limit_property</td>
<td>Data Types</td>
</tr>
<tr>
<td>Lower_limit_value</td>
<td>double</td>
<td>The lower limit of a value range.</td>
<td>Value_range_property</td>
<td>Data Types</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class</td>
<td>UoF</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Lower_offset_value</td>
<td>double</td>
<td>The lower limit defined as the lower offset value from the single value (value + lower limit). Source: ISO 10303:239 Product Life Cycle Support.</td>
<td>Value_with_tolerances_property</td>
<td>Data Types</td>
</tr>
<tr>
<td>Month_component</td>
<td>int</td>
<td>The month element of the Date expressed as a value between 1 and 12, where: 1 = January 2 = February 3 = March 4 = April 5 = May 6 = June 7 = July 8 = August 9 = September 10 = October 11 = November 12 = December. Source: ISO 10303:239 Product Life Cycle Support.</td>
<td>Date</td>
<td>Data Types</td>
</tr>
<tr>
<td>Property_creation_date</td>
<td>Date</td>
<td>The date when the property value was determined.</td>
<td>Property_representation</td>
<td>Data Types</td>
</tr>
<tr>
<td>Property_determination</td>
<td>Class</td>
<td>The method by which the value of the property has been determined. The value of this attribute need not be specified. Where applicable the following classes must be used: - ‘Calculated’: the value has been calculated, - ‘Designed’: the value represents a value intended by the design, - ‘Estimated’: the value has been estimated, - ‘Measured’: the value has been measured, - ‘Set point’: the value is used as an initialization value. Source: ISO 10303:239 Product Life Cycle Support.</td>
<td>Property_representation</td>
<td>Data Types</td>
</tr>
<tr>
<td>Text</td>
<td>char</td>
<td>The string that is the element of representation.</td>
<td>Text_property</td>
<td>Data Types</td>
</tr>
<tr>
<td>Data element</td>
<td>Type</td>
<td>Definition</td>
<td>Class</td>
<td>UoF</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Unit</td>
<td>Class</td>
<td>The unit with which the quantity is expressed.</td>
<td>Value_with_unit_property</td>
<td>Data Types</td>
</tr>
<tr>
<td>Upper_limit_value</td>
<td>double</td>
<td>The upper limit of a value range.</td>
<td>Value_range_property</td>
<td>Data Types</td>
</tr>
<tr>
<td>Upper_offset_value</td>
<td>double</td>
<td>The upper limit defined as the upper offset value from the single value (value + upper offset value).</td>
<td>Value_with_tolerances_property</td>
<td>Data Types</td>
</tr>
<tr>
<td>Value</td>
<td>double</td>
<td>The value of the quantity.</td>
<td>Value_with_unit_property</td>
<td>Data Types</td>
</tr>
<tr>
<td>Year_component</td>
<td>int</td>
<td>The year element of the Date expressed as a value between 1 and 9999.</td>
<td>Date</td>
<td>Data Types</td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td>End of data module</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>