International specification for training analysis and design

S6000T-B6865-06000-00

Issue No. 2.2

Publishers:

AeroSpace, Security and Defence Industries Association of Europe

Aerospace Industries Association of America (AIA)

Copyright, user agreement and special usage rights

1 Copyright

All correspondence and queries should be directed to:

ASD
Rue du Trône
1050 Brussels
Belgium

2 Agreement for use of the specification S6000T™ suite of information

2.1 Definitions

S6000T™ suite of information means, but is not limited to:

- the training needs analysis specification S6000T
- examples (eg, XML instances, PDF files, stylesheets), XML schemas
- the input specification S6000X
- any other software or information on the page titled "S6000T Downloads" on the website www.s6000t.org

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

2.2 Notice to user

By using all or any portion of **S6000T™ 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 **S6000T™** 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 Holder grant to you a non-exclusive license to use **S6000T™ suite of information.**

2.4 Intellectual property rights

S6000T™ 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 **S6000T™ suite of information** and all rights not expressly granted are reserved by the Copyright Holder.

2.5 No modifications

You shall not modify, adapt or translate, in whole or in part, **S6000T™ suite of information**. You may however add business rules.

2.6 No warranty

S6000T™ 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 **S6000T™ suite of information**. The Copyright Holders make 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 Holders 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 **S6000T™ 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 **S6000T™ 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 ICC by three arbitrators appointed in accordance with the said Rules of Arbitration. All related proceedings should be at the location of the ICC in Paris, France. The language to be used in the arbitral proceedings shall be English.

3 Special usage rights

Permission to use, reference, or deliver training from the information contained in this document, and in the S-Series IPS Specifications, and the right to reproduce or publish the S-Series IPS Specifications, in whole or in part, is hereby given to the following:

1. National Associations who are members of ASD and all their member companies.
2. Members of Aerospace Industries Association of America.
3. Armed Forces that are customers of Companies included in these special usage rights Categories 1 and 2 inclusively.
4. Ministries and Departments of Defense of the member countries of ASD, AIA, NATO, and NATO Partners.

5 NATO bodies, organizations & agencies.

6 The industries of ASD, AIA, NATO’s nations, and NATO’s Partners.

7 Universities, Educational Institution, Technologies and Research Institutes within ASD, AIA, NATO nations and NATO Partners.

Highlights

This issue includes updates to the Copyright, user agreement and special usage rights only. There have been no technical changes in this issue. The highlights below are those of the 2021 Block Release.

List of tables

Page

Table 1 References

Chap No./Document No.	Title
Chap 1	Introduction to the specification
Chap 2	Information gathering
Chap 3	Analysis
Chap 4	Design
Chap 5	In-Service Human Performance Optimization
Chap 6	Relationship to other specifications
Chap 7	Data model
Chap 8	Terms, abbreviations and acronyms

Table 2 General

Chapters	Summary of changes
All	There are several major changes in this issue of S6000T including data model changes All Chapters have had editorial changes.

Table 3 Chapter 1 - Introduction to the specification

Chapters	Summary of changes
All	Editorial changes

Table 4 Chapter 2 - Information gathering

Chapters	Summary of changes
All	Editorial changes

Table 5 Chapter 3 - Analysis

Chapters	Summary of changes
Chap 3.3.4	Detailed description of Training Gaps is provided, and process described to identify training gaps and record them.
Chap 3.3.3	Addition of identifying Performance Objects as task are identified
Chap 3.5	Deletion of identifying Performance Objectives from this process

Table 6 Chapter 4 - Design

Chapters	Summary of changes
Chap 4.1 thru Chap 4.3	Editorial changes.
Chap 4.4	Addition of training gaps process outputs as input to the learning objectives design process, and update to process steps for writing LO statements and developing supporting LO statements to account for associations to training gaps.
Chap 4.5 thru Chap 4.12	Editorial changes.

Table 7 Chapter 5 - In-Service Human Performance Optimization

Chapters	Summary of changes
All	Editorial changes

Table 8 Chapter 6 - Relationship to other specifications

Chapters	Summary of changes
All	Editorial changes

Table 9 Chapter 7 - Data model

Chapters	Summary of changes
All	Editorial changes **Note** The following chapters also had technical changes.
Chap 7.3.6	Additions to the data model

Chapters	Summary of changes
Chap 7.3.14	Additions to the data model
Chap 7.3.22	Additions to the data model
Chap 7.3.31	Additions to the data model
Chap 7.3.33	Additions to the data model
Chap 7.3.40	New Chapter introducing Training Gaps.
Chap 7.3.44	Additions to the data model

Table 10 Chapter 8 - Terms, abbreviations and acronyms

Chapters	Summary of changes
Chap 8.1	Existing definitions are amended because of other changes brought about by this issue of S6000T
Chap 8.2	Existing acronyms are amended because of other changes brought about by this issue of S6000T

Table of contents

Chap No.	Title		Applic
Chap 6.2	Relationship to other specifications - S2000M	S6000T-A-06-02-0000-00A-040A-A	All
Chap 6.3	Relationship to other specifications - S3000L	S6000T-A-06-03-0000-00A-040A-A	All
Chap 6.4	Relationship to other specifications - S4000P	S6000T-A-06-04-0000-00A-040A-A	All
Chap 6.5	Relationship to other specifications - S5000F	S6000T-A-06-05-0000-00A-040A-A	All
Chap 6.6	Relationship to other specifications - SX000i	S6000T-A-06-06-0000-00A-040A-A	All
Chap 7	Data model	S6000T-A-07-00-0000-00A-009A-A	All
Chap 7.1	Data model - General	S6000T-A-07-01-0000-00A-040A-A	All
Chap 7.2	Data model - Overview	S6000T-A-07-02-0000-00A-040A-A	All
Chap 7.3	Data model - Units of functionality	S6000T-A-07-03-0000-00A-040A-A	All
Chap 7.3.1	Units of functionality - UoF Aggregated Element	S6000T-A-07-03-1000-00A-040A-A	All
Chap 7.3.2	Units of functionality – UoF Applicability Statement	S6000T-A-07-03-2000-00A-040A-A	All
Chap 7.3.3	Units of functionality - UoF Breakdown structure	S6000T-A-07-03-3000-00A-040A-A	All
Chap 7.3.4	Units of functionality - UoF Change Information	S6000T-A-07-03-4000-00A-040A-A	All
Chap 7.3.5	Units of functionality - UoF Circuit Breaker	S6000T-A-07-03-0500-00A-040A-A	All
Chap 7.3.6	Units of functionality - UoF Competency Definition	S6000T-A-07-03-0600-00A-040A-A	All
Chap 7.3.7	Units of functionality - UoF Course	S6000T-A-07-03-0700-00A-040A-A	All
Chap 7.3.8	Units of functionality - UoF Course Element	S6000T-A-07-03-0800-00A-040A-A	All
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan	S6000T-A-07-03-0900-00A-040A-A	All
Chap 7.3.10	Units of functionality - UoF Digital File	S6000T-A-07-03-1000-00A-040A-A	All
Chap 7.3.11	Units of functionality - UoF Document	S6000T-A-07-03-1100-00A-040A-A	All
Chap 7.3.12	Units of functionality - UoF Environment Definition	S6000T-A-07-03-1200-00A-040A-A	All
Chap 7.3.13	Units of functionality - UoF Hardware Element	S6000T-A-07-03-1300-00A-040A-A	All

End of data module

Chapter 1

Introduction to the specification

Table of contents

Chapter 1.1

Purpose

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
S3000L	International procedure specification for Logistics Support Analysis (LSA)

1 Purpose

Successful training support is based on detailed, comprehensive analysis and solid design definition data. This data provides a firm foundation for planning, decision making, and execution and support of Product training. Therefore, the purpose of the S6000T is to define all levels of analysis and design to deliver relevant and effective Product training.

2 Background

The concept of this specification was originated in the Aerospace Industries Association of America (AIA) and the AeroSpace and Defence Industries Association of Europe (ASD). Prior to that, procedures for producing training information were planned to be a chapter in S3000L. However, the importance of Product training needs analysis using, amongst other things, outputs from S3000L was deemed to warrant a separate specification.

Note

The term Integrated Logistics Support (ILS) has been replaced throughout this specification with Integrated Product Support (IPS).

Chapter 1.2

Scope

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
S1000D	International specification for technical publications using a common source database

1 General

S6000T is derived from the Analysis, Design, Development, Implementation, and Evaluation (ADDIE) model for instructional systems design.

2 Scope

The scope of S6000T includes aspects of the ADDIE model and other models harmonized as a single model. However, for this initial issue, the scope is limited to information gathering, analysis, and design.

Note

S6000T is in the field of Integrated Product Support (IPS) as is the rest of the S-Series IPS specifications.

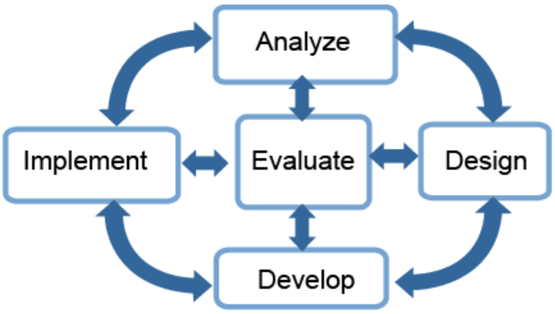
The S-Series IPS specifications have adopted a concept of a Product, which is defined as:

– A Product is a representation of a family of items which share the same underlying design purpose. For example:
  - Airbus A340
  - Aegis Class Destroyer

- Stryker
- Ford Fusion
- Pegasus Engine
- iPhone 7

S6000T also uses this concept in its processes to deliver its training support products.

S6000T adheres to the Instructional Systems Design (ISD) process. The ISD systematic process involves the assessment and development of training solutions, designed specifically for the purpose of formal training delivery, using primarily the ADDIE model, which is the most widely recognized ISD model. The ADDIE model is made up of the five phases depicted in its name. Refer to Fig 1.



ICN-S6000T-B6865-00001-001-01

Fig 1 ADDIE model

ADDIE is an acronym for the five phases of this model to training development:

- Analysis Phase - The analysis phase determines what must be trained. The first step is to determine whether a training problem exists and then to identify possible solutions. A sequence of processes and analytical models are then used to identify critical human tasks and to identify the standards, conditions, performance measures, and other criteria needed to perform each task.
- Design phase - The instructional design is based on the analysis phase results. In this phase, the instructional designers also develop Learning Objectives (LO), an assessment strategy, and test items, as well as design the instruction. The instructional strategies are also developed in this phase and instructional methods and media are selected. The output of the design phase is the Training curriculum.
- Development phase - Instructional development is based on the design phase results. During the development phase, lesson materials, unit exercises, drills, and other instructional materials for both the student and the instructor are developed. Media, selected in the design phase, is produced during this phase.
- Implementation phase - After the instructional system has been designed and developed, and the validation activities of formative and summative evaluation have been completed, the instructional system is implemented. In this phase, the instructional system is fielded.
- Evaluation Phase - Evaluation is a continuous process that starts during the analysis phase and continues throughout the development and life cycle of the instructional system and the Product. After the implementation phase, periodic internal and external evaluation provides feedback that is used to ensure the continuing accuracy and quality of the instructional system. This process is referred to as In-Service Human Performance Optimization (ISHPO). Refer to Chap 5.

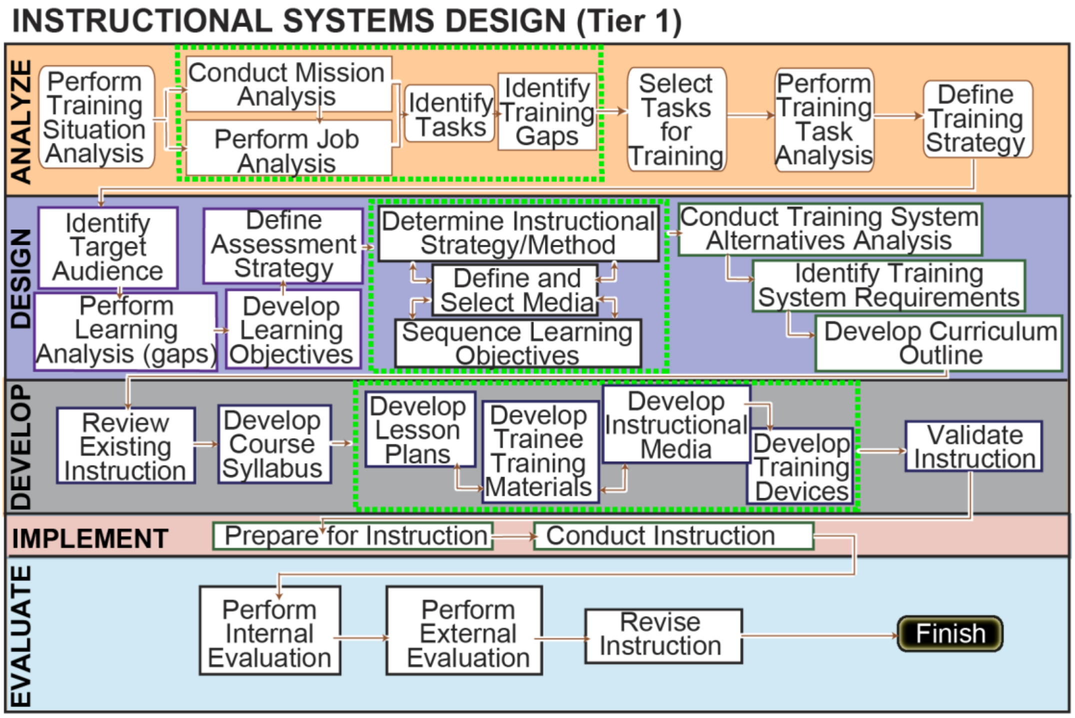
The goal of the ISD process is to optimize the training Return on Investment (ROI) by increasing the effectiveness of education and training. Examples of the products of ISD are:

- instructional systems based on mission, job, and performance requirements
- courses consisting of relevant knowledge and skills instruction

– graduates having the necessary Knowledge, Skills, and Attitudes (KSA) to perform the mission, job and/or task

S6000T follows a process-based architecture approach which is a systematic process whereby a process is broken down into lower-level tiers to identify a series of actions or operations. It begins with a Tier 0 process flow which is a high-level flow as provided in Fig 2.

A Tier 0 process flow is decomposed into a Tier 1 which indicates a series of actions or operations. Refer to Fig 2, Chap 3 and Chap 4 further decompose each of the Analysis and Design actions into Tier 2 process steps with identified inputs and outputs of each step. The outputs represent a high-level look at the data that results from these steps. These objects form the basis for the Data Model that is provided in Chap 7.



ICN- S6000T-B6865-00002-003-01

Fig 2 Instructional Systems Design (Tier 1)

Chapter 1.3

How to use the specification

Table of contents

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 1	Introduction to the specification
Chap 1.1	Purpose
Chap 1.2	Scope
Chap 1.4	How to tailor S6000T
Chap 1.5	Maintenance of the specification
Chap 2	Information gathering
Chap 2.1	Information gathering - General
Chap 2.2	Information gathering - Interviewing
Chap 2.3	Information gathering - Existing source material
Chap 2.4	Information gathering - Knowledge and experience
Chap 3	Analysis

1 General

To assist the reader in navigating this specification, each of the S6000T chapters are described in this chapter.

2 Field of application

S6000T is an international specification that is designed to cover the processes for delivering training information in support of any Product. The aspects of training information that are covered include:

- information gathering
- analysis
- design

3 Basic definitions

the Product	A family of items which share the same underlying design purpose.
Product variant	A member of a Product family which is configured for a specific purpose and is made available to the market.
Project	The overall set of Integrated Product Support (IPS) activities defined for a Product.
Training Support Products (TSP)	The results of the activities described in S6000T (eg, learning gap analysis report, training strategy, etc)

4 Reading conventions

4.1 General

Throughout S6000T specific conventions are used to aid common understanding and to minimize duplication. These conventions are:

Mandatory	Required by S6000T or any other of the S-Series IPS specifications
Optional	Aspects of by S6000T or any other of the S-Series IPS specifications used at the discretion of the user
training support products	The results of the activities described in S6000T (eg, gap analysis report, training strategy, etc)

5 Organization of S6000T

S6000T is organized into 8 main chapters.

5.1 Chapter 1

Chap 1 introduces S6000T, and gives:

- the purpose. Refer to Chap 1.1.
- the scope. Refer to Chap 1.2.
- details on how it can be tailored. Refer to Chap 1.4.
- how to request a change to the specification. Refer to Chap 1.5.

5.2 Chapter 2

The procedures for gathering information in general are given in Chap 2 together with:

- general information on gathering. Refer to Chap 2.1.
- gathering information by interviewing. Refer to Chap 2.2.
- gathering information by using existing information. Refer to Chap 2.3.
- gathering information by using knowledge and experience. Refer to Chap 2.4.

5.3 Chapter 3

Chap 3 describes the analysis process, and provides:

- general introduction to the analysis processes used in S6000T Refer to Chap 3.1.
- Training situation analysis - identify the need for training and possible solutions. Refer to Chap 3.2.
- mission and job - identifying a unit's mission, all the specified, implied, and supporting missions and all the collective tasks that must be performed to accomplish each of these missions. Decomposing, structuring, and describing jobs to establish training requirements. Refer to Chap 3.3.
- task selection - analysis of the identified tasks to determine which need training. Refer to Chap 3.4.
- task analysis - breaking a skill down into smaller, more manageable components to determine the required knowledge, skills, and attitudes. Refer to Chap 3.5.
- training strategies - identify the strategy for achieving the required knowledge, skills and attitudes. Refer to Chap 3.6.
- business objects - Listings of the total data elements related to S6000T and those that are unique to S6000T. Refer to Chap 3.7.

5.4 Chapter 4

Chap 4 gives details of the process for designing new training courses and lessons and gives:

- general introduction for the design phase procedures and processes used in S6000T. Refer to Chap 4.1.
- target audience description - collecting and analyzing information on a target population, then using analysis results to create a training plan. Refer to Chap 4.2.
- learning gaps analysis - to actual performance with desired performance in terms of knowledge, skills and attitudes. Refer to Chap 4.3.
- learning objectives - identifying what students should know as a result of the training. Refer to Chap 4.4.
- assessment strategy – identifies the requirements for measuring and assessing the target audience's attainment of LO's. Refer to Chap 4.5.
- instructional strategy Refer to Chap 4.6.
- media Selection – determines the most efficient and effective ways to deliver learning content to trainees. Refer to Chap 4.7.
- learning objective sequencing - collecting learning objectives together and then ordering them in an effective learning sequence. Refer to Chap 4.8.
- training system alternative - a structured process for analyzing alternative training solutions to determine the best training solution based on a program's constraints. Refer to Chap 4.9.
- training system requirements - describes all the resource requirements for the development and delivery of a curriculum outline. Refer to Chap 4.10.
- curriculum outline approval and release - describes the process for placing the product training curriculum outline under version control and formalizing it as part of the training system. Refer to Chap 4.11.
- business objects - Listings of the design phase data elements related to S6000T. Refer to Chap 4.12.

5.5 Chapter 5

Chap 5 describes the process for optimizing in-service human performance, including:

- the In-Service Human Performance Optimization (ISHPO) - a general description of the process. Refer to Chap 5.1.
- ISHPO preparation for optimizing human performance - examines if an analytical basis for the valid training support is still relevant, correct and up to date. Refer to Chap 5.2.
- ISHPO analysis for optimizing human performance - detailed analysis of the product training and its associated technical publications and logistics. Refer Chap 5.3.

- ISHPO follow up phase guides the training analyst through the various training changes and activities that were identified during the analysis phase. Refer to Chap 5.4.

5.6 Chapter 6

Chap 6 describes the relationships between S6000T and other specifications, including the general relationship with the other S-Series IPS specifications. Refer to Chap 6.1.

5.7 Chapter 7

Chap 7 describes the S6000T data model.

5.8 Chapter 8

Chap 8 is the glossary of terms and list of abbreviations and acronyms.

Chapter 1.4

How to tailor S6000T

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 1.4.1	How to tailor S6000T - Project specific tailoring
Chap 1.4.2	How to tailor S6000T - Conformance and compliance

1 General

An implementation of S6000T should be tailored to produce the deliverable for a specific project. For contracting purposes, the data produced using S6000T must conform to the rules for the S-Series data model and the processes should comply with the processes described in S6000T.

Chap 1.4.1 introduces the concept of project specific tailoring of S6000T. Chap 1.4.2 defines the requirements necessary to fulfill to stay in adherence with the specification, in terms of conformance and compliance.

Chapter 1.4.1

How to tailor S6000T - Project specific tailoring

Table of contents

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 3.2	Analysis - Training situation analysis
Chap 3.6	Analysis - Training strategy
Chap 3.7	Analysis - Business objects
Chap 4.2	Design - Target audience description
Chap 4.11	Design - Curriculum outline
Chap 4.12	Design - Business objects
S6000X	Input data specification for S6000T
SX002D	Common data model for the S-Series IPS specifications

1 General

S6000T has been produced to cater for many different types of Products. Therefore, to make it suitable for a given project, some aspects of tailoring will be required. It is recommended that the tailored version of this specification is referred to in the project's contractual documentation.

1.1 Common data model

Any tailoring must not affect the Common Data Model (CDM) data types declared and described in SX002D, which are:

- primitives
- compound attributes
- business objects
- classes and attributes included in the S6000T Units of Functionality (UoF)

1.2 S6000T data model

Depending on the required deliverable, select only the business objects and data elements that serve as inputs to the production of the deliverable for the respective analysis defined in Chap 3.2 thru Chap 3.6 and Chap 4.2 thru Chap 4.11. The relationship between the respective analysis and the data model is defined in Chap 3.7 and Chap 4.12. These business objects and data elements are identified by S6000X.

2 Project business rules

Project business rules give the rules and guidance for the use of the optional data elements and UoF that are needed to produce the tailored deliverable. These business rules must be agreed between parties to document the details of the agreed tailoring of this specification. These rules must cover the requirements for optional elements, their population from specific data sources, and the use of specific values.

3 Units of functionality

The UoF used by S6000T are:

- UoF Aggregated Element
- UoF Applicability Statement
- UoF Breakdown Structure
- UoF Change Information
- UoF Circuit Breaker
- UoF Competency Definition
- UoF Course
- UoF Course Element
- UoF Curriculum and Course Plan
- UoF Digital File
- UoF Document
- UoF Environment Definition
- UoF Hardware Element
- UoF Job Duty
- UoF Learning Assessment
- UoF Learning Gap
- UoF Learning Objective
- UoF Message
- UoF Mission Definition
- UoF Organization
- UoF Part Definition
- UoF Performance Parameter
- UoF Product
- UoF Product Usage Context
- UoF Product Usage Phase
- UoF Remark
- UoF Resource Specification
- UoF Security Classification
- UoF Software Element
- UoF Subtask Train Prioritization
- UoF Target Audience
- UoF Task
- UoF Task Knowledge Skill and Attitude
- UoF Task Performance Objective
- UoF Task Resource
- UoF Task Target

- UoF Task Train Prioritization
- UoF Training Analysis and Design Message Content
- UoF Training Entry Requirement
- UoF Training Gap
- UoF Training Media and Fidelity
- UoF Training Media Resource
- UoF Training System Resource Requirement
- UoF Warning Caution Train Prioritization
- UoF Zone Element

Chapter 1.4.2

How to tailor S6000T - Conformance and compliance

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 1.4.1	How to tailor S6000T - Project specific tailoring
Chap 3	Analysis
Chap 4	Design
Chap 5	In-Service Human Performance Optimization
SX002D	Common data model for the S-Series IPS Specifications

1 General

Implementing S6000T for a project involves tailoring of the specification. In making such tailoring it is necessary to know the type and degree of tailoring that is allowable and acceptable without leaving the S6000T concept. The criteria described in this chapter do not necessarily extend into future issues.

2 Conformance and compliance

2.1 Background and basics

The degree of adherence to the data, process and tools specified by S6000T is expressed in terms of compliance and conformance to the specification.

2.2 Certification

It should be noted that, even though this specification defines what S6000T conformance and compliance means, there is no official certification for any kind of S6000T training support products, processes, software, or implementations.

2.3 Definitions

2.3.1 Data

Data, or a set of related data, is conformant to S6000T if and only if it fulfills the form requirements of such data, as specified by S6000T.

2.3.2 Process

A process and/or a procedure is compliant with S6000T if, and only if, it is carried out in accordance with the processes and procedures specified by S6000T and does not violate the criterion in Para 2.3.3.

2.3.3 Conformant/conformance

Conformant and conformance refer to the data specified in S6000T. The rules for the data are fixed. Refer to SX002D. Therefore, conformance does not depend on any tailoring.

2.3.4 Compliant/compliance

Compliant and compliance refer to the adherence to one or more process or procedure as defined by S6000T. Compliance depends on the required deliverable. If a deliverable (refer to Chap 1.4.1) is tailored out, then the process that would have produced that deliverable is not required and therefore cannot be adhered to.

2.4 Criteria

There are two forms of S6000T criteria:

- core criteria
- optional criteria

The S6000T features, for which conformance/compliance criteria can be defined, are:

- general, overall properties
- the analysis processes
- the Common Data Model (CDM) and the S6000T specific data model and the elements they contain

2.4.1 Core criteria

S6000T provides rules, processes and procedures covering various aspects of the training needs analysis process.

Some requirements concerning these aspects can be crucial in obtaining commonly comprehensible S6000T information. These requirements form a core of criteria for S6000T conformance and compliance.

All aspects that the specification declare as mandatory and all aspects that have a "must" statement are core criteria.

Core criteria is defined as criteria that must be fulfilled to achieve S6000T conformance and compliance. Refer to Para 2.3.

2.4.2 General core criteria

S6000T features are S6000T Core conformant and related processes are S6000T Core compliant if and only if they meet the following criteria.

2.4.2.1 Data model

To be S6000T conformant with regards to the data model, all mandatory elements in the CDM must be used in accordance with SX002D.

2.4.2.2 Analysis processes

To be S6000T analysis process compliant an analysis process must, at a minimum, fulfill the requirements described in Chap 3, Chap 4 and Chap 5 (including subchapters) in accordance with the layered business rules and the rules given at Para 2.4.1.

2.4.3 Optional criteria

In addition to the core criteria, there are optional aspects that users of the specification can apply at their discretion. Optional criteria are defined as criteria that are set by a project using the optional features in S6000T. The criteria below all relate to functional aspects of S6000T which are optional.

2.4.3.1 Optional features

It is possible to be S6000T compliant/conformant while choosing not to be compliant/conformant regarding optional features described in the specification. However, to reach S6000T conformance and/or compliance in any of the listed aspects the criteria for the chosen feature must be fulfilled.

Aspects not explicitly covered by the specification do not influence the degree of conformance and compliance. Further, where there is a decision point identified in the specification, projects can do whatever is within the bounds of the decision point and still be compliant/conformant.

2.4.3.2 Optional processes

A project can decide whether to use all or part of an optional process. However, the project must take into account the conformance/compliance aspects of all inputs to the whole or part process.

Chapter 1.5

Maintenance of the specification

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
ILSC-2018-001	Governance of the S-Series IPS specifications

1 Maintenance of the specification

S6000T is maintained by the S6000T Steering Committee (SC) operating under the supervision of the Integrated Product Support Council. Both the Council and this SC include representatives from AeroSpace and Defense Industries Association of Europe (ASD) and Aerospace Industries Association of America (AIA) member companies and nations.

Note

The term Integrated Logistics Support (ILS) has been replaced throughout this specification with Integrated Product Support (IPS).

Issues related to S6000T can be raised using the change request tool found at www.sx000i.org. Change requests are submitted with the understanding that any revisions to S6000T can affect the other specifications in the S-Series IPS specifications, and that proposed changes are subject to international agreement between ASD and AIA member companies and nations.

Upon receipt of a change request, the S6000T SC will follow the change management process described in ILSC-2018-001, to gain consensus agreement from the participating organizations prior to the publication of changes. The S6000T SC considers change proposals and ratifies them for incorporation into S6000T.

Chapter 2

Information gathering

Table of contents

Chapter 2.1

Information gathering - General

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 2.2	Information gathering - Interviewing
Chap 2.3	Information gathering - Existing source material
Chap 2.4	Information gathering - Knowledge and experience

1 General

To determine the training requirements, information must be gathered and evaluated. Ideally, the information obtained will enable a well-defined, accurate, and complete description of the current situation regarding competencies of the workforce. This chapter provides guidance to be used in Information gathering throughout the Instructional System Design (ISD) process.

2 Description

Several sources of information can be used in the ISD process. Availability of data will most likely vary for each training project and the sources of information will be different. For example, in the early stage of a new product development, final technical documentation is not always available, so other sources can be required. Other examples that can lead to availability of data and different sources of information include:

- developing initial training requirements for a new system
- identifying training requirements to accommodate a major modification to an existing system
- updating a training system due to changes in mission or operational requirements.
- if an existing training system will be evaluated and analyzed

Rarely will all information be available especially early on in a project. For that reason, many techniques can be required to gather the necessary training requirements. The key is to be able to focus on the information needed to determine training requirements quickly and not have to read thousands of pages of information. Training analysts must plan their information requirements and techniques before beginning data collection. The training analyst should decide what information and which sources will be necessary for the project. The analyst should then design eg, a data collection sheet for use during data collection to ensure that all necessary data is collected. The data collection sheets should be discussed between all involved parties that are going to collect the data to ensure that the data is collected in the same manner. More than one method should be used since each method has its inherent strengths and weaknesses.

Three traditional methods of gaining information are:

- interviewing key stakeholders including SMEs. Refer to Chap 2.2.
- observation of existing material and information. Refer to Chap 2.3.
- reports from previous knowledge and experience. Refer to Chap 2.4.

Chapter 2.2

Information gathering - Interviewing

Table of contents

List of tables

References

Table 1 References

Chap No./Document No.	Title
None	

1 General

Interviewing is an efficient, flexible, and rewarding way of gathering information from stakeholders. For example, semi-structured interviews are particularly valuable because they allow for follow-up questions of interviewees when clarification about a response is needed or if a new aspect needs to be determined.

2 Description

Observation, research, and conducting interviews are the primary tools of the analyst. The interview is a specific form of a come together (eg, meeting or conference), and is usually limited to two persons, the interviewer, and the interviewee. In special circumstances there can be more than one interviewer or interviewee in attendance. In these cases, there should still be one primary interviewer and one primary interviewee.

Interviewing is a basic method to gather information regarding interviewees' knowledge, expectations, behaviors, feelings, preferences, and modes of thought and reason. The data collected can be subjective and not necessarily based on facts. Interviewing can be both qualitative and quantitative depending on the layout. The interviewing skills of the analyst determine what information is gathered, the quality, and depth of the information. Because interviewing is a primary technique for information gathering during the training requirements phase of a project is a skill which should be mastered by every analyst.

3 Method

Interviewing is a versatile method because of the many various techniques for data gathering. There are several methods of interviews each having their own advantages. The training analyst should consider the information they are trying to collect and the people they are interviewing when choosing the most appropriate method.

3.1 Unstructured interview

An unstructured interview consists of open and random questions and is best suited when the interviewer has some key questions formulated in advance but only has a vague perception of what information is relevant. This allows the interviewer to ask questions based on the interviewee's responses to get qualitative data. The interviewee can speak freely about their opinions thereby leading the interviewer in the desired direction. Since each interviewee can be asked a different series of questions, this style lacks the reliability and precision of a structured interview.

3.2 Structured interview

A structured interview has a fixed format. When conducting a structured interview, the interviewer uses a predefined template in which all questions are prepared beforehand and are put in the same order to each interviewee. The answers can still be open or predetermined based on options like yes/no or on a scale. Although this style lacks the free flow of a friendly conversation (as in an unstructured interview) it provides the precision and reliability required in certain situations. This will give quantitative data that is easier to analyze than the answers from an unstructured interview. A structured interview will require that the interviewer has good insight in the subject and a clear picture of what to investigate.

3.3 Semi-structured interview

A semi-structured interview is a mixture of unstructured and structured interviews that takes a flexible approach and permits questions to arise in response to the dialog. It is also the most used option. In this case, the template to be used consists of a predefined structure of which subjects to be addressed. The interviewer can decide the order of the questions and ask supplementary questions, also known as probing. Probing is a method where supplementary questions are used to get deeper knowledge of the answer. For example, “Can you develop that?”, “What do you mean by that?” etc. This method enables a more relaxed interview environment and deeper understanding of the subject matters.

3.4 Group interviews

In the group interview, interviewees are assembled to give information relative to their job. The training analyst asks questions about job performance and can ask the group to list data on tasks that cannot easily be demonstrated or observed. Because the group interview involves recall rather than recognition it can provide inaccurate or incomplete data.

4 Conducting the interview

A good interview starts with simple and non-specific questions. Begin by explaining the purpose of the interview. Explain how the interviewee’s data and any data collected will be used from the interview. Try to keep leading questions to a minimum.

First, decide on what type of interview method is appropriate.

Second, create an interview template based on the interview method using the examples in Table 2.

Note 1

For unstructured and jury-of-expert interview, the key questions should be defined beforehand

Note 2

For the observation it is important to focus on the left-hand column in Table 2 which will help with asking the interviewee the correct questions.

Third, when collecting data, focus on getting the necessary information for the training analysis. This can be done by asking the right question during information gathering. Examples are given in Table 2.

Table 2 Asking the right question

To identify	Ask the question
Procedures, activities, steps	What does the person do first? Next?
Tools, materials	What is used to perform the task?
Product Information	What system do you use to perform the task?
Cues	How do you know when to perform what?
Work environment or conditions	Under what conditions is the task performed
Performance Standards	What is the Standard of acceptable performance?
Use of Technology	What technology or equipment do you need to perform this task
Goals or Objectives	What is the goal of the task?
Examples of workarounds	When a problem arises, what sort of workarounds are normally performed to complete the task?
Regulations	Which regulations and/or specific law must be followed?
Prerequisite	What must be available to start the task?

Chapter 2.3

Information gathering - Existing source material

Table of contents

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 2.4	Information Gathering Knowledge
S3000L	International procedure specification for Logistic Support Analysis (LSA)
S4000P	International specification for developing and continuously improving preventive maintenance
SX000i	International guide for the use of the S-Series of Integrated Product Support (IPS) specifications.

1 General

Existing source material is a critical means of gathering information regarding the training needs of a product. Available material will vary depending on the nature of a product. For a new product, or a major change to a product there can be limited material available. Training analysis should use multiple sources identified here to get accurate information to determine training requirements.

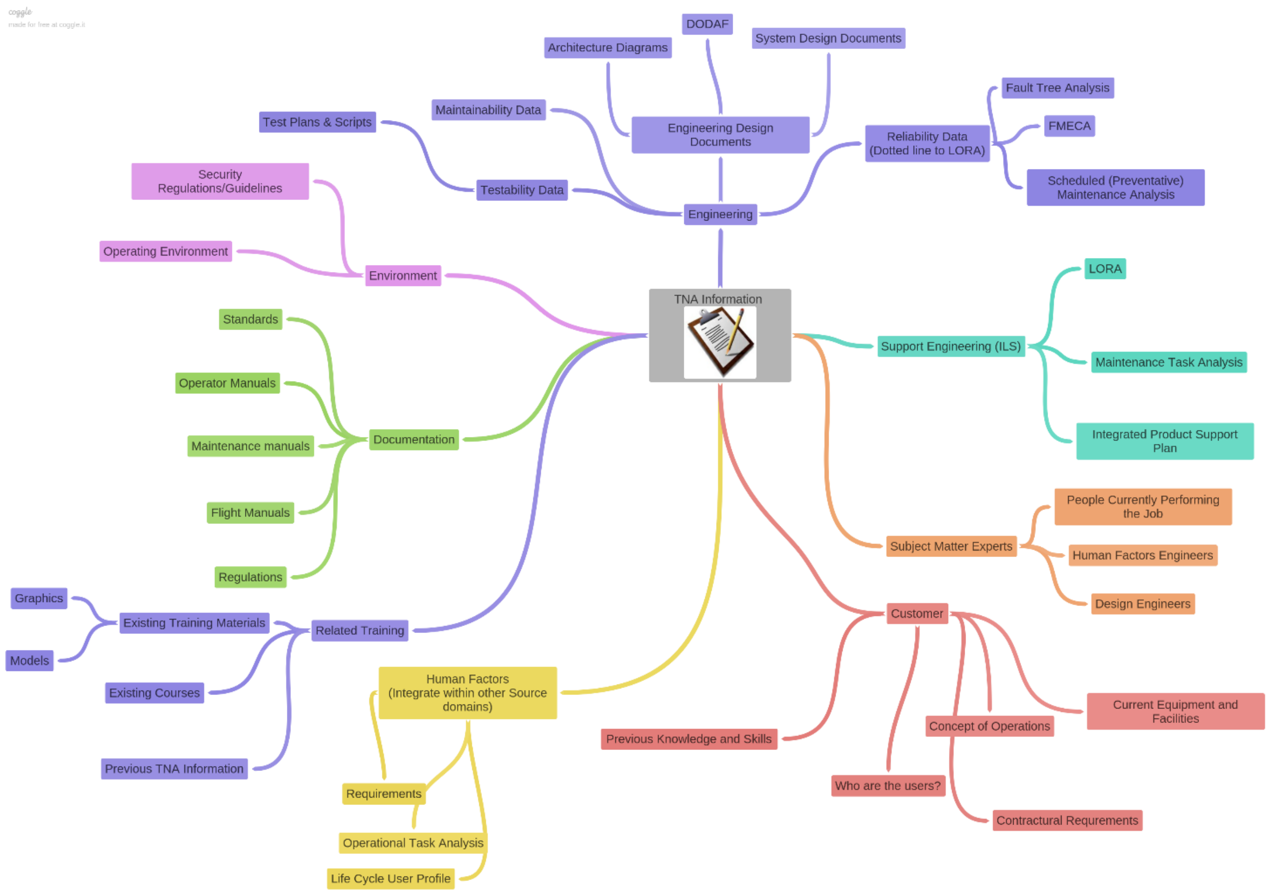
2 Description

There are many sources of material that can help in the establishment of training requirements, and the development of a training curriculum outline for a product. These sources can be in various states of maturity based on the life cycle of the product. Training analysts should work with the customer as well as with other product domains such as engineering, manufacturing, and product support disciplines to ensure the latest documents are available.

For information to be useful there must be confidence in its reliability as well as its validity. Reliability is the collection of repeatable results with subsequent measurements of the same thing. With reliable measurements, confidence can be gained in what is observed is the true situation. Validity proves that the information is accurate and is what it is supposed to be. Testing validity can be a complex process especially when looking at variables that can differ from person to person. However, with reliable measurements and testing validity, confidence can be gained in what is being observed is the true situation.

3 Sources

There are multiple sources of material that can provide valuable information when conducting a Training Needs Analysis (TNA). The sources used depend largely on the type of information needed to conduct a TNA for a particular Product. These sources are illustrated in Fig 1.



ICN S6000T-B6865-00003-001-01

Fig 1 Information sources

Training analysts should use as many of these sources as possible to ensure accurate information is obtained.

3.1 Engineering design documents

Engineering design documents are a critical source of information for TNA. In the case of new product development or a major product change these documents can be one of the few sources of information. There are numerous engineering documents that can be used. The most frequently used include:

- architecture diagrams
- system design documents
- test plans and scripts
- maintainability data
- testability data
- reliability data including Failure Mode, Effects and Criticality Analysis (FMECA) data and other fault tree data

3.2 Environment

Environmental data including operating environment information, and security regulations guidelines should be reviewed carefully for TNA related information. This information can provide critical information for conditions in which a product is operated and maintained.

3.3 Documentation

Technical documentation including operating, maintenance, and flight manuals as well as regulations and standards provide critical user and maintenance task information as well as task condition and standard information. Technical documentation for new product development can be sparse, so Training analysts should coordinate closely with their technical data departments to ensure information can be made available as quickly as possible.

3.4 Related training programs

Training programs similar in nature to the new environment including comparable or related products should be reviewed for training requirements relevant to the current product. This information includes training material (course, and TNA), student population, training equipment, and other training environment factors. This information is critical especially when conducting a Training Situation Analysis (TSA).

3.5 Human factors engineering

Human Factors Engineering (HFE) focuses on the human's role to operate and maintain the product in two major categories, user analysis and usability analysis. In user analysis the attention is on the usage objective of the system, the environment where the system will be used, and variables such as the user roles, user characteristics, and the user interface requirements. In usability analysis the emphasis is on the complexity of the system, and the improvements that can be made to the design of the system to enhance its usability. This information is critical especially when determining user tasks.

3.6 Customer

Training analysts should work closely with the customer when conducting a TNA. Customer information including performance requirements, contractual requirements, training goals, equipment, facilities, and organizations as well as user information provide important data and facts for determining training requirements.

3.7 Subject matter experts

Subject Matter Experts (SME) are a valuable source of information for determining training requirements. For more information regarding how to collect information from SMEs. Refer to Chap 2.4.

3.8 Support engineering

Training analysts should coordinate closely with other product support disciplines when determining training requirements. Level of Repair Analysis (LORA) and Maintenance Task Analysis (MTA) are a critical source of information, Refer to S3000L. In addition, Preventive maintenance requirements as well as the Integrated Product Support Plan are additional critical sources. Refer to S4000P, and SX000i.

4 Collecting the information

When collecting data, focus on getting the necessary information for training analysis. As indicated there can be numerous sources of information available especially for a mature product, so identifying the right source is critical for gathering the information quickly. This can be done by mapping the source to the information required. Refer to Table 2.

Table 2 Finding the right source

To identify	Information sources
Procedures, activities, steps	Engineering, HFE, SME, Product support engineering, and technical documentation
Tools, materials	Customer, engineering, environment, HFE, Product support engineering
Product information	Customer, engineering, technical documentation
Cues	Environment, HFE, SME, technical documentation
Work environment or conditions	Customer, environment, SME, environment
Performance standards	Customer, SME, support engineering
Use of technology	Customer, engineering, HFE, related training
Goals or objectives	Customer, engineering, Product support, related training
Examples of workarounds	Engineering, HFE, Product support, engineering, SME,
Regulations	Customer, Product support engineering
Prerequisite	Customer, related training, SME

Chapter 2.4

Information gathering - Knowledge and experience

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 2.2	Information gathering - Interviewing

1 General

Information that is already known and drawn from previous experiences can be very useful for gathering information. Individuals who have this knowledge are identified as Subject Matter Experts (SME). SMEs are any individual who have:

- experience performing the job or task being analyzed
- knowledge of the subject matter
- experience with current or similar training systems
- knowledge/experience mastering the performance requirements of a job/task

These individuals can provide critical information and insight when determining the training requirements of a new or revised Product.

2 Description

There are several different methods to obtain information from SMEs. Each of these methods have advantages as well as disadvantages so choosing the right method to use depends largely on the current situation. As with other information sources, it is often advantageous to use multiple methods depending on the requirements.

3 Method

3.1 Interview

Interviewing SME's is a critical method for collecting information. There are several different types of interviews. For guidance on conducting interviews refer to Chap 2.2.

3.2 Observation

The observation/interview method involves sending the instructional analyst to observe and interview SMEs and their supervisors on the job. Observing the SMEs at work during their interactions with products and/or services in their natural surroundings allows flexibility in gathering the required data by providing the training analyst opportunities to continually evaluate the information obtained. Direct observation of personnel as they perform their job, combined with interviews, provides the most useful source of task information. The instructional analyst should have a thorough understanding of the literature and functional relationships of the job to correctly interpret and describe the behaviors observed.

Observation can be direct or unobtrusive:

- direct observation, which is when the subject knows that somebody is watching and evaluating his performance. There are two types of direct observations:
  - continuous monitoring, which is monitoring through behavior or body language of the individual.
  - time allocation, which allows a researcher to randomly allocate selecting a place and certain time of observation.
- unobtrusive observation, which is when the subject does not know he/she is being observed during the interview.
  - the bias of a user stating what he/she thinks is wanted to be heard is avoided and real-time specific insights from user behavior can also be gained although this type of observation is limited to a point in time

However, observations do not always fit with every project, but it is insightful to witness how users perform their tasks

3.3 Jury-of-experts

In the jury-of-experts method, experienced and knowledgeable personnel from various activities are brought together to record and analyze the data on jobs for which many critical behaviors are not directly observable. This method can effectively supplement on-site observation/job analysis and written surveys to gain perspective from other people in the organization and occasionally those outside the organization. The experts are selected for their experience and knowledge of the job.

Chapter 3

Analysis

Table of contents

Chapter 3.1

Analysis - General

Table of contents

Page

List of tables

List of figures

References

Table 1 References

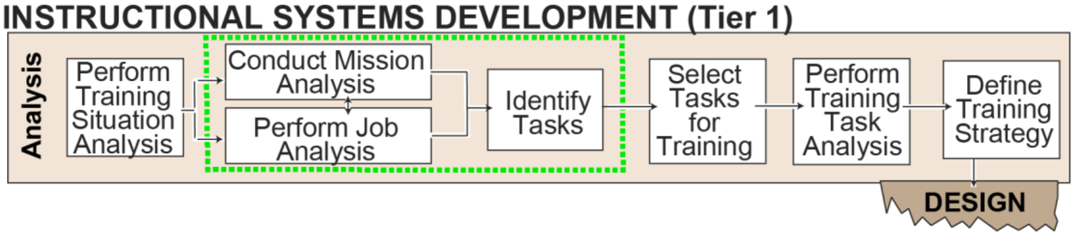
Chap No./Document No.	Title
Chap 3.2	Analysis - Training situation analysis
Chap 3.3.1	Mission, job, and task analysis - Mission analysis
Chap 3.3.2	Mission, job, and task analysis - Job analysis
Chap 3.3.3	Mission, job, and task analysis - Identify tasks
Chap 3.3.4	Mission, job, and task analysis – Training gaps
Chap 3.4	Analysis - Task selection
Chap 3.5	Analysis - Task analysis
Chap 3.6	Analysis - Training strategy
Chap 3.7	Analysis - Business objects
S3000L	International procedure specification for Logistic Support Analysis (LSA)

1 General

The analyze phase determines what must be trained. A result of the analyze phase is the identification of Knowledge, Skills, and Attitude (KSA) required for mission/job/task performance.

2 Description

Training needs analysis is a systematic process of understanding training requirements. It is required to determine whether a need for training exists and whether an intervention will contribute to the achievement of the Product's support requirements. The steps of the analysis phase are shown in Fig 1.



ICN-S6000T-B6865-00004-004-01

Fig 1 Analysis phase

The first step in this phase is to perform a Training Situation Analysis (TSA) to determine if a training problem exists and to identify possible solutions. Refer to Chap 3.2.

A sequence of processes and analytical models are then used to identify critical human tasks and to identify the standards, conditions, performance measures, and other criteria needed to perform each task. For training that ties its content directly to preparing trainees for the performance of a mission or job, the mission/job performance requirements are analyzed, and a set of task inventories are developed if they are not available from other sources, (eg, S3000L).These processes are described in:

- mission analysis. Refer to Chap 3.3.1.
- job analysis. Refer to Chap 3.3.2.
- identify tasks. Refer to Chap 3.3.3.
- identify training gaps. Refer to Chap 3.3.4.

These task Inventories are then analyzed to determine training prioritization. This process is task selection. Refer to Chap 3.4.

Once tasks are identified as to whether they require training, and their associated training priority, KSA required for task performance are determined. This is referred to as task analysis. Refer to Chap 3.5.

Having conducted the analysis, a training strategy can be established. Refer to Chap 3.6.

The S-Series IPS specifications use a database, described by a data model, to capture the results of the analysis and the training strategy in business objects. Refer to Chap 3.7.

Chapter 3.2

Analysis - Training situation analysis

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.2.1	Training situation analysis - Example TSAR outline
Chap 5.1	In-service training support optimization - General

1 General

Training Situation Analysis (TSA) is performed to define the need for training and to identify and evaluate possible alternative solutions and is normally performed very early on in a Product's lifecycle. It is often performed as part of Product support analysis activities.

2 Description

The TSA takes a broad look at all aspects of an existing or emerging training situation or program to determine if there is a training need or other human performance solutions required. The TSA analyzes system performance and supportability requirements and defines the training or other human performance needs to meet these requirements. The TSA results in a recommended strategy for meeting all identified needs, including rough order of magnitude cost estimates, milestones, and schedule. If the recommended strategy includes a training system, then concept exploration activities are initiated to define the desired training system.

The initiation of a TSA can originate from a variety of sources. Generally, they will consist of a perceived need:

- for new or additional training due to a new design or new Product
- to influence a Product design change
- to resolve performance deficiencies in a current design

The TSA should be completed as early as possible and prior to any other analysis, design, or development activity. If performed effectively, a TSA will:

- identify assumptions, constraints, opportunities, and risks within a training requirement
- identify programming, resourcing, or operating issues of existing or proposed training systems, within the boundaries of policy
- identify any non-training issues within a requirement
- provide options to resolve the requirement to meet stakeholder needs and expectations

An effective TSA is the structured, logical, and diligent analysis of the source requirement, using a range of relevant supporting data. This requires the identification and collation of relevant documents, as well as the identification of appropriate stakeholders and subsequent data collation, through properly structured interview questions.

The output of the analysis will be a Training Situation Analysis Report (TSAR), which will provide a range of appropriate options to resolve the source requirement. The results of the TSAR should be clearly evidenced, with reasoned arguments and recommendations and be fully auditable from the source requirement, supporting data and TSA results.

A TSAR can provide evidence that the source requirement is not a training issue; for example, it can indicate the need for:

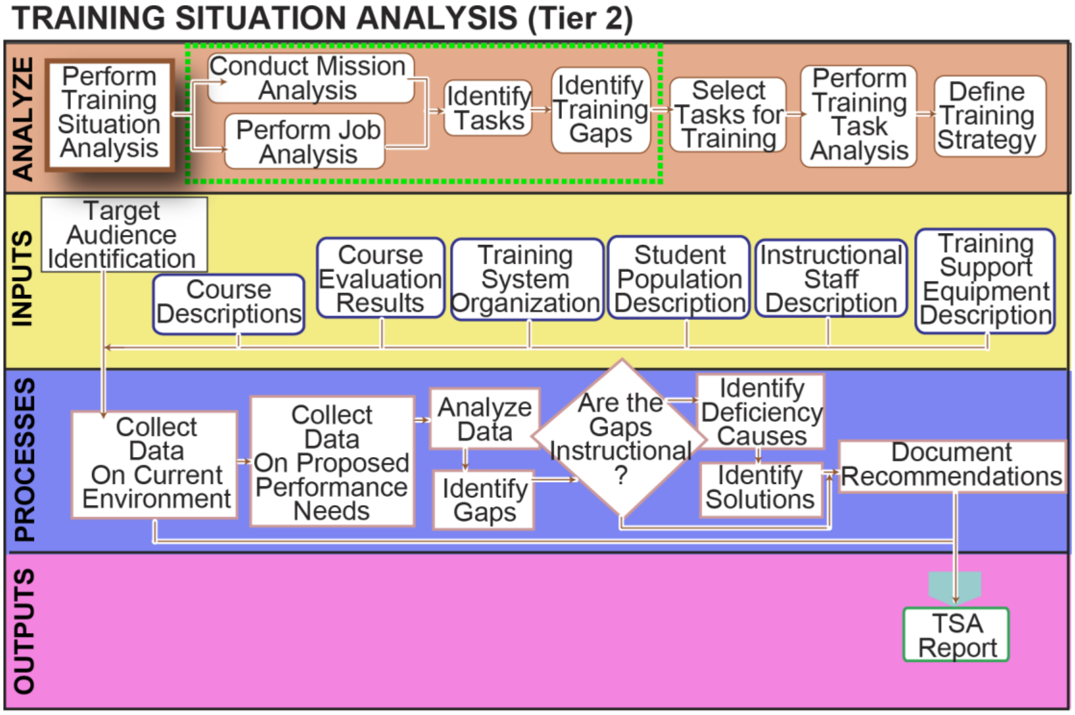
- a revision of procedures
- re-allocation of tasks
- procurement of equipment
- improvements to management and supervision

If a training solution is not recommended, further analysis, design and development need not proceed. However, where a training solution is selected, an outline recommendation and rationale for further analysis, design, and development should be included.

To meet the scope and complexity of a TSA, it is highly recommended that an appropriately qualified and experienced Training Analyst conducts the TSA. TSAs require experience in data collection and interviewing, analysis, and reporting, as well as a thorough understanding of training.

3 Process

Fig 1 shows the TSA process.



ICN-S6000T-B6865-00005-002-01

Fig 1 Training situation analysis

3.1 Inputs

The inputs for the training situation analysis process include:

- Target audience identification, refer to Para 3.1.1
- Course descriptions, refer to Para 3.1.2
- Course evaluation results, refer to Para 3.1.3
- Training system organization, refer to Para 3.1.4
- Training audience description, refer to Para 3.1.5
- Instructional staff description, refer to Para 3.1.6
- Training support equipment description, refer to Para 3.1.7

3.1.1 Target audience identification

This is the identification of the audience who will be operating and maintaining the product. It includes how trainees will be selected to take part in the training for the new or updated product. It includes training audience sources, anticipated attrition rates, current or anticipated class, etc. This includes how trainees will be chosen for the new training system.

3.1.2 Course descriptions

This includes a description of existing similar relevant training courses for similar systems, or earlier versions of the same system. It also includes a description of training equipment (simulators, Interactive Courseware (ICW), etc) used for each lesson or module of the current system.

3.1.3 Course evaluation results

This includes any course evaluations, trainee's surveys, etc, used as part of the existing similar relevant training system. These will be used to inform the TSAR of lessons learned.

3.1.4 Training system organization

Identifying the training organization and their roles and responsibilities is important for determining who has important existing training situation information and providing essential contact information for performing subsequent training development activities. The training system organization should include operational and training doctrines and practices that will used.

3.1.5 Trainee audience description

A description of the anticipated trainee audience is critical for determining the breadth and depth of training requirements. This description should include a description of their anticipated training background, as well their current job experiences. Anticipated training roadmap(s) for all potential trainees should be identified if possible.

3.1.6 Instructional staff description

Understanding the breadth and detail of training staff that will be available to support and maintain the proposed training solution is critical to help refine training recommendations. Support staff includes, but is not limited to:

- instructors
- training equipment
- engineers
- program managers
- administrators
- courseware support personnel

This description should include an organizational structure of the training staff, so roles and responsibilities are clearly identified. This description should also include initial training staff requirements, as well as recurrent training staffing requirements throughout the lifecycle of the training.

3.1.7 Training support equipment description

A description of anticipated training equipment is also critical for determining training requirements. Training equipment includes, but is not limited to:

- simulators
- trainers
- classrooms
- computers
- support equipment

It is important to provide as much detail about this equipment as possible including, numbers, availability, etc.

3.2 Process steps

The training situation analysis process steps are:

- Collect data on current environment, refer to Para 3.2.1
- Collect data on proposed performance needs, refer to Para 3.2.2
- Analyze data, refer to Para 3.2.3
- Identify gaps, refer to Para 3.2.4
- Identify deficiency causes, refer to Para 3.2.5
- Identify solutions for gaps and deficiencies, refer to Para 3.2.6
- Document recommendations, refer to Para 3.2.7

3.2.1 Collect data on current environment

The current environment includes anything that is similar to the new environment. For example, if the new Product is a replacement, a modification, or an upgrade to an existing Product, then data is gathered for the Product. Similarly, any Product that is similar in performance or mission requirements to the new Product is worthy of having information gathered about it for comparison. This data includes all the data inputs mentioned in Para 3.1.

3.2.2 Collect data on proposed performance needs

Performance needs of the new system, including mission/business goals, maintenance plans, operating parameters, logistical requirements, and high-level training goals, etc, of the new system, are gathered during this step in the process.

3.2.3 Analyze data

The data collected from the current Product, as well as the performance needs of the new Product should be analyzed. First, what is working from the current Product should be documented. Also, areas of improvement needed in the current Product should also be identified. Refer to Chap 5.1. Also, the performance requirements should be analyzed. Similarities and differences between the requirements of the new Product compared to the old Product should be documented.

3.2.4 Identify gaps

During this step, the data about current systems is compared with the information on the new systems. Gaps between these two systems are identified and documented. These gaps are documented in terms of situational statements. Situational statements describe single or multiple situations/events/occurrences which are having either positive or negative effects upon the ability of the organization to perform its mission.

3.2.5 Identify deficiency causes

During this step, deficiencies are identified in terms of human performance on the current training system. These deficiencies are written as impact statements that must include a full description of the specific resource deficiencies, excesses caused by the situation and their impacts on mission performance. The impact statements must identify those resource areas that have been affected by the situation and must include:

- personnel
- instructors
- administrative personnel
- support personnel
- trainees
- training simulators and other support equipment
- facilities
- training material

3.2.6 Identify solutions for gaps and deficiencies

During this step, solutions and alternative approaches are identified that can address these gaps and these deficiencies. These solutions and alternatives must describe how to realistically reduce or resolve the impact upon the ability of an organization to perform its mission. Each alternative should be described in terms of what will be done to reduce or resolve the deficiencies. These solutions should identify resources, including cost estimates required to complete these implementations. These solutions must cover all human performance solutions, not just training solutions. Solutions such as better documentation should also be considered.

3.2.7 Document recommendations

The results of this analysis are documented in a TSAR. The TSAR should clearly articulate the objectives and the scope of the analysis as well as any assumptions identified in starting the analysis. A high-level description of the approach taken to conduct the analysis should be included.

The current situation, including similar programs should be described in detail, as well as the methodology used to identify these similar programs. Training equipment, including classrooms, simulators, etc, of the current or similar systems, should be clearly described. An architectural representation of how all the training equipment ties together is also recommended. A detailed description of the current trainee audience, including how they are chosen for this training, should be described. Attrition rates, including reasons for the attrition, should be identified. If available, the latest curriculum of the current or similar systems should be included for reference purposes.

The new or planned system should be described in as much detail as possible, including any assumptions of the planned user audience, (eg, skills, background, etc). The findings and recommendations, including human performance recommendations and cost estimates, should be clearly identified.

3.3 Outputs

The output of the TSA is documented in the TSAR. This report is intended for program managers to help assess the high-level training requirements for a new or revised program as well as set the stage for a program guidance conference. The report sections should include, but are not limited to:

- identified recommendations and prioritization of recommendations
- alternatives as applicable
- cost estimates
- major milestones and schedules
- identified gap alternatives, including non-training alternatives
- recommended intervals of recurrent or refresher training

The results of the TSA should be discussed in a guidance conference with the customer to review documented gaps, the possible solutions to those gaps and to identify high level training requirements.

3.4 TSAR data outline

An example TSAR data outline is given at Chap 3.2.1.

Chapter 3.2.1

Training situation analysis - Example TSAR outline

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
None	

1 General

The Training Situation Analysis Report (TSAR) documents the findings of the Training Situation Analysis (TSA). It evaluates existing training system and assesses the impact of the training system on the current training. Another part of the TSA provides the means to identify relevant new training technology and to examine the applicability of new technology that meet the training requirements of the new system.

Note

This outline is a recommendation only and can be modified as required.

The TSA data will provide a description of the analysis performed to verify the efficiency of the training system to meet existing training needs. It will also survey other similar training programs for applicability to new training needs and must include:

1.1 Introduction

The introduction will contain an overview of the purpose and expected application of the data and the results compiled from this analysis. It should describe how training capabilities and system requirements, identified from this analysis, will influence the eventual design, development, and operation of the training system.

1.2 Scope

The scope of the study.

1.3 Assumptions

Any assumptions that the study is based on and any restrictions of the study.

1.4 Methodology

The methodology should be a high-level description of the approach the training analyst will use. It is important to define and document all processes before beginning analysis and development. The methodology should include the analyst's description of taking a task thread through the entire model.

1.5 Data Sources

A list of major data sources used in conducting the study.

2 Existing situation

2.1 Training program mission

Document the mission statement(s) of the training programs that comprise the existing training system.

2.2 Historical background

A description of the historical background of the existing training situation.

2.3 Training program organization

Identifying the training organization, together with its roles and responsibilities, is important for determining who has relevant existing training situation information and providing essential contact information for performing subsequent training development.

2.4 Curriculum

A general description of existing curriculum for each training phase or course, to include, but not be limited to:

– Major goals, content, length, and integration into curriculum

- Each phase, with sub-sections describing academic, synthetic, operational equipment and practical job training instructional aids
- Performance measurement methods and applications
- Feedback and evaluation procedures

2.5 Student population

Data relating to student population in terms of:

- Entry level requirements and noted exceptions
- Personnel qualification standard(s)
- Student Population source(s)
- Attrition rates
- Current class size and range of anticipated size
- Prerequisite deficiencies, to include learning deficiencies or problem levels including reading readiness, academic background, and aptitude

2.6 Training equipment materials and documents

A description of training equipment and materials presently used in the program of instruction. The description should include:

- Training simulators, to include types, locations, numbers, capability assessment, deficiencies and planned modifications and procurement
- Operational equipment, to include types, number available, utilization rates and availability
- Major weapon systems, to include availability in terms of training requirements and schedules
- Instructional media, to include types, count, capability, configuration, reliability, and maintainability
- Instructional materials, to include types, subject content, authoring language, capability, authoring systems, version, and adequacy in terms of curriculum requirements
- Media support capabilities, to include facilities and maintenance in terms of specific categories of media
- Command support, to include photography, printing, graphics, quality, timeliness, coordination requirements and funding requirements

2.7 Training facilities

A brief description of each facility presently used for the conduct of training programs at all sites. For each facility, the description should contain:

- The demand (use) for a specific facility
- The condition of that facility. Illustrations, both interior and exterior, must be included if available
- The optimal condition of the facility, to adequately accommodate the current and anticipated demand (eg, human factors and adherence to safety, fire, and environmental regulations, etc)
- Facility improvements required to meet the optimal condition
- Training ranges, to include types, capabilities, utilization, and planned modifications
- A description of financial support, including the sources, types of money and authorizations used in the operation of the organization
- Facility limitations including environmental conditions, power requirements, etc

3 Situational analysis

The situational analysis will provide a detailed description of those situation(s) which affect the ability of the organization to perform its mission. The situation analysis should include:

3.1 Analytical methods and procedures

The analytical methods and procedures data will provide a description of the methods, procedures, data compilation and analytical tools used to accomplish this analysis and derive the system requirements for the training system. It should contain detailed flow diagrams of the process and describe each discrete event step of the process. Any assumptions made during the analytical process should be described and rationale provided.

3.2 Situational statement

The situational statement will describe single or multiple situations/events/occurrences which are having either positive or negative effects upon the ability of the organization to perform its mission. The situational statement should also include a description of any lessons learned.

3.3 Impact statements

The impact statements will include a full description of the specific resource deficiencies, excesses caused by the situation and their impacts on mission performance. For example, it is possible that one situation can impact one or more of the following resources or that multiple situations can affect only one of the following resources. The impact statements should identify those resource areas that have been affected by the situation and must include, but not be limited to:

- Personnel (eg, military, civilian, contractor, etc) including, but not limited to:
  - Instructors
  - Administrative personnel
  - Support personnel (eg, operation, maintenance, etc)
  - Students
- Training equipment
- Facilities including:
  - Instructional
  - Support
- Training material
- Specific mission impacts

3.4 Literature review

The literature review will include the findings of any documents related to the situation, impacts, alternatives, and recommendations.

3.5 Information sources and data collection

The information sources and data will include a description of all information sources and data collected during this analysis. It should include all locations, organizations and personnel contacted and used as information sources. It should also describe the objective, utilization and expected result for each informational event.

3.6 Solutions and alternatives

The solutions and alternatives data will describe those alternatives which can realistically reduce or resolve the impact upon the ability of an organization to perform its mission. Each alternative should be described in terms of:

- What must be done to reduce or resolve the deficiency
- What resources are needed
- Predicted effectiveness
- Associated life cycle costs
- Time to complete implementation of the alternative

The alternative should consider those instances where a resource is affected by more than one situation. For example, if there are two distinct situations which affect the organization's instructor personnel, it is possible that one alternative can resolve both situations. The solutions and alternatives should include functional diagrams, plus descriptions of the hierarchy of requirements and show the relationship between program, system and training requirements at all levels.

3.7 Recommendations

The recommendations should identify the alternative selected as being most efficient and cost-effective. Cost data should include any savings in resources that will result from adopting the recommendations. Each recommendation should include, but not be limited to:

- A description and depiction of the organizational support and training concept recommended for the training system under consideration
- A description of the relationships among:
  - Concepts
  - Functional capabilities
  - Characteristics of training devices
  - Media and system requirements
- For clarification, a block and flow relationships among:
  - Concepts
  - Functional capabilities
  - Characteristics of training diagrams for clarification
- A description of the system level requirements for the chosen training system conceptual design
- Justification for selecting that recommendation
- A description of resources required to implement the recommendation, including, but not limited to:
  - Personnel quantity and type
  - Funding amount and type
  - Other applicable resources
  - Return On Investment (ROI)
  - Milestones for implementing the recommendations
  - The name of the principal action agency responsible for the recommendation
  - The names of the agencies responsible for providing support to the principal action agency
- A description of the ramifications of not implementing the recommended alternative

3.8 Summary

The summary will include a synopsis of all recommendations. It should include brief statements highlighting each of the items described in the recommendations. A plan of action and milestones for the recommendations should also be included. This plan of action and milestones should include initial training as well as the sustainment of training system.

4 Training technology assessment

The training technology assessment data contains information on technology used to provide training for existing systems, which are similar to the emerging system for which training must be developed. It also provides an assessment of the training technologies that are used in areas relevant to the training requirements of the emerging system.

4.1 Similar system analysis

The similar systems analysis will include the results, assessments and recommendations made from this analysis for each training program under consideration. It should also include a matrix and description that compares training programs and depicts similarities and differences among programs for selected training features (eg, simulators, computer-based instruction, etc).

4.2 Associated training

The associated training data will identify all major training programs similar to the training program under development.

4.3 Commonality analysis

The commonality analysis data will contain a description of the results of the commonality analysis performed on the training equipment identified and characterized in the associated training equipment listed above. This information should include, but not be limited to:

- The optimal number and mix of training equipment required to support the training under study
- A description of the optimal simulation features for each type of equipment
- A description of the optimal instructional features for each type of equipment
- A description of common features that fail to support training, as well as common causes for failure
- An estimate of criticality for each feature, for each type of equipment

4.4 State-of-the-art assessment

The state-of-the-art assessment data addresses the training program under consideration. It should include separate descriptions for results, assessments, and recommendations in the areas of training concepts, methods, techniques, technology, and system evaluation. It should address the instructional and evaluative attributes of each area. It should describe alternative and optimum combinations of training concepts, methods, techniques, and technologies with rationale for the training system under consideration. Results of the assessment presented should include, but not be limited to:

- Identification of technology which will be upgraded
- Identification of relevant technologies which could be applicable to the training program
- An estimate of criticality for each identified technology
- A Return on Investment of each identified technology

4.5 Simulation and instructional features

The simulation and instructional features data will provide summary information on the simulation and instructional features identified in commonality and state-of-the-art assessment data. Refer to Para 4.4.

End of data module

Chapter 3.3

Analysis - Mission, job, and task analysis

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 3.3.1	Mission, job, and task analysis - Mission analysis
Chap 3.3.2	Mission, job, and task analysis - Job analysis
Chap 3.3.3	Mission, job, and task analysis - Identify tasks
Chap 3.3.4	Mission, job, and task analysis - Training gaps
S3000L	International procedure specification for Logistic Support Analysis (LSA)

1 General

The mission, job and task analyses define human performance requirements and facilitates their achievement with a system design process. This process integrates traditional training analysis with human systems integration, systems engineering, and Product support analysis to ensure users and maintainer's needs are aligned and traced to mission and system design requirements.

2 Product

Understanding the Product and its systems, functions, as well as its hardware and software components, is core to all training analysis and design activities. It is also important that training analysis and design have a common understanding of the Product as other Integrated Product Support (IPS) disciplines. The way that the Product structure is organized for training analysis and design must therefore be based on the same principles as described in S3000L to ensure consistent representations across all S-Series IPS specifications.

A Product in terms of the S-Series IPS specifications is defined as a family of items, which share the same underlying design purpose.

Examples

- Nexter VBMR Griffon
- Boeing, F-18
- Aegis Class Destroyer
- Ford Fusion

A Product then comes in one or many variants where each Product variant is configured for a specific purpose and is made available to the market.

Product structures and change management ensures the correct identification of different Product configurations, controls changes, and records the change implementation of the physical, functional and system characteristics of the Product's structure, systems, subsystems, equipment, and components.

Establishment of a Product breakdown is essential to identify support Product structure for all configurations during the project life cycle (refer to S3000L). Traceability between design structure and support breakdown structure is needed to control possible changes in the design Product structure during all Project phases and implement those modifications into the support breakdown.

These Product structures and Product breakdowns are normally established during the Logistical Support Analysis (LSA), refer to S3000L. S6000T must, where applicable, use these same Product structures and Product breakdowns so that training requirements can be associated to same Product and Product variant, and ensure the digital thread back to LSA and Product design. This enables training analysis and training design activities to immediately react on changes upstream in timely and correct manner.

3 Analysis

Analysis is the foundation that determines the requirements of any new or major change to a Product. The goal of mission, job, and task analysis is to identify a set of task inventories per job/role that detail the technical tasks that are used to operate or maintain a Product. This chapter first establishes the structure of the Product that training must support, while integrating all training analysis activities. It should be stressed that training is only one of several disciplines that conduct similar analysis to determine a set of Task Inventories. Logistics Support Analysis, (LSA) and Human Factors Engineering (HFE) use similar processes to identify similar task inventories. It is recommended that training analysts coordinate with these other disciplines to identify a single set of task inventories. The processes outlined herein should only be used if training analysis is being performed as a standalone support element. Refer to S3000L.

Mission analysis utilizes a Top-Down Functional Analysis (TDFA) to identify human system integration requirement methodology to develop an audit trail that ties mission (operational) requirements to training requirements to human activities. Refer to Chap 3.3.1.

Job analysis provides a methodology to detail a list of tasks required to perform a specific job or duty in support of a set of mission requirements. Refer to Chap 3.3.2.

Task analysis provides a methodology to identify and document collective and individual tasks required to support mission or operational requirements. Refer to Chap 3.3.3.

Training gaps provides a process for identifying and documenting training gaps generated by the new or changed product. Refer to Chap 3.3.4.

Chapter 3.3.1

Mission, job, and task analysis - Mission analysis

Table of contents

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.7	Analysis – Data element list
S3000L	International procedure specification for Logistic Support Analysis (LSA)

1 General

A mission is a pre-established objective or purpose for a military or business unit.

2 Description

A mission (or operational) analysis is systematic study to identify:

- a unit's mission
- all the specified, implied, and supporting missions that the unit and its subordinate units should perform as part of that overall mission
- all the mission tasks that must be performed to accomplish each of these missions

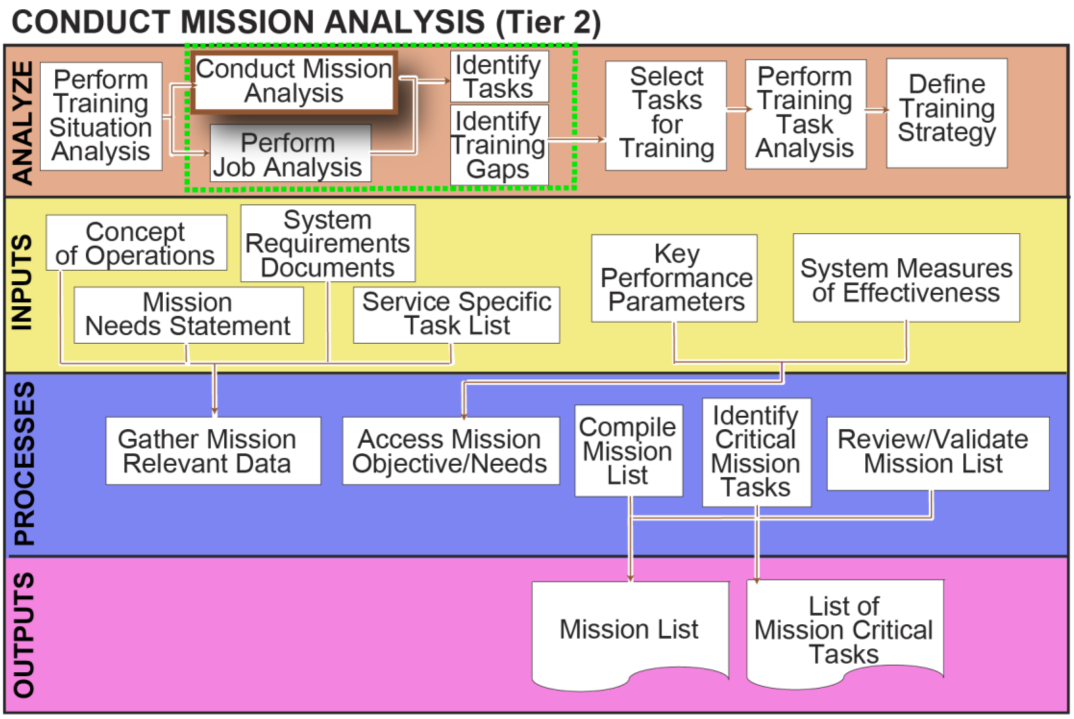
Mission analysis is conducted on new and existing Products and normally result from the following conditions:

- creation of new end-item
- significant changes in the operational concept and or usage of an existing end-item
- changes in the mission or operational capability of a Product

First the Product is analyzed into mission or operational objectives, goals, and activities in a logical sequence. Next, it is necessary to identify the functions needed to satisfy the mission or operational objectives. Functions are described as activities performed by a Product to satisfy the requirements of the mission. Finally, tasks describe what humans do to functionally operate and maintain the product to support all aspects of a mission.

3 Process

The mission analysis process is shown at Fig 1.



ICN-S6000T-B6865-00005-004-02

Fig 1 Mission analysis process

3.1 Inputs

The inputs for the mission analysis process are:

- concept of operations, refer to Para 3.1.1
- mission needs statement, refer to Para 3.1.2
- system requirements documents, refer to Para 3.1.3

- service specific task lists, refer to Para 3.1.4
- key Performance parameters, refer to Para 3.1.5
- system measures of effectiveness, refer to Para 3.1.6
- engineering design information, refer to Para 3.1.7
- human/system interface views, refer to Para 3.1.8
- human performance requirements, refer to Para 3.1.9
- stakeholder, subject matter experts, refer to Para 3.1.10

3.1.1 Concept of operations

Concept of operations are defined in a series of documents that define the scope of the Product and its boundaries. They define the overall objectives of the Product as well as supporting descriptions and requirements. Examples of these documents are Initial Capabilities Documents (ICD) and Performance Based Specification (PBS).

3.1.2 Mission (operational) needs statement

The operational need statement identifies the overall purpose of the Product. It declares the need that the Product going to fulfill in an overall operational environment.

3.1.3 System requirements documents

The system requirements documents identify the overall requirements of the Product. These documents identify any Product constraints that can affect human performance and thus identify a training impact. These requirements documents identify crew size as well as facility or environmental constraints.

3.1.4 Service specific task lists

Service specific task lists, also known as organizational task lists, identify jobs or roles that operate or maintain sections of the overall Product, or the Product itself. They can also identify and job or role restrictions in terms of required qualifications.

3.1.5 Key performance parameters

Key Performance Parameters (KPP) specify what the critical performance goals and performance measures are for a Product. KPPs indicate critical performance success measures for a Product and help determine critical training goals. Examples of KPPs critical to mission analysis include survivability KPPs such as speed and maneuverability, material availability, operational availability, and Training readiness.

3.1.6 System measures of effectiveness

In addition to KPP’s additional measures of effectiveness are also critical in determining mission or operational performance. These Measures of Effectiveness (MOEs) are a measure of a Product’s ability to support the achievement of an operation mission. They are defined in terms of operational results and will drive the performance measurement in the overall training curriculum.

3.1.7 Engineering design information

Engineering design information is used to provide functional views of a Product. This information includes architectural views, conceptual views, and preliminary and detailed design drawings of individual parts the Product. This information represents design requirements of the Product and helps identify human/system interface views.

3.1.8 Human/system interface views

Human system interface views provide information on the human’s role in the operation and maintenance of a Product. This includes information on user analysis as well as user ability analysis. User analysis focusses on the usage objective of the Product, the environment where the Product will be used and variables such as the user roles, user characteristics, and any user interface requirements. Usability analysis focusses on the complexity of the Product, and the Product design to identify opportunities to improve its usability.

3.1.9 Human performance requirements

Human performance requirements document the measures of the performance of tasks on a Product. Human performance is defined as the ability of a user or maintainer to accomplish a task requirement. These performance measures are used to determine the training performance requirements required in a task.

3.1.10 Stakeholder subject matter experts

Subject Matter Experts (SME), who are performing the same or similar jobs on similar equipment, are a good source of information for determining operational or mission requirements and for detailing human interaction requirements of a Product.

3.2 Process steps

The mission analysis process steps are:

- gather mission relevant data, refer to Para 3.2.1
- assess mission objective needs, refer to Para 3.2.2
- compile a mission list, refer to Para 3.2.3
- identify critical mission tasks, refer to Para 3.2.4
- review and validate mission tasks, refer to Para 3.2.5

3.2.1 Gather mission-relevant data

The first step in the process is to gather all relevant data regarding the mission or operational performance. This includes the documentation and resources that guide, direct, or explain the activities of the target organization. Key performance parameters as well as other system MOEs should be identified during this step.

3.2.2 Assess mission objective needs

During this step, mission relevant data and the other inputs identified at Para 3.1, are analyzed to determine the overall purposes, objective, or capabilities of the Product and to identify and document the mission or operational requirements. Training analysts must first determine the operational need for the Product by reviewing the information gathered in at Para 3.2.1. Training analysis must also document any constraints that could affect human performance, and have a impact on training. The training analyst must consider the infrastructure and all associated logistics associated with the Product. This includes identifying minimum or maximum crew size, and facility or environment constrains that could impact the training system.

3.2.3 Compile a mission list

During this step, the Products primary and secondary mission or operational requirements are analyzed, and the results are documented. This includes interviewing individuals who are familiar with high-level operational requirements of the Product and the type of anticipated/planned operational scenarios in which the Product will be involved. MOE and Measures of Performance (MOP) are reviewed so that student performance requirements can be determined.

Functional analysis is also performed during this step. The function analysis involves determining what the Product is designed to do to satisfy the mission requirements. For example, if an operational requirement of the Product will be able to stop within ten feet, then it is up to the Product engineer to ensure the Product has a braking system. The training analyst must concentrate on the activities required for successful operation and maintenance of this braking system. Functional analysis involves an identification and analysis of the:

- identification of all the necessary functions required to satisfy the primary and secondary mission or operational scenarios of the Product
- enumeration of the performance measures for each function
- operational relationships between the functions. For example, relationships between the wheel system, and braking system on the Product must be analyzed to ensure performance requirements can be met.

3.2.4 Identify the critical mission tasks

Task analysis is performed to obtain information that is used to develop tasks that describe how humans will perform Product functions. Use of Product requirement documents, such as architectural views and system functional traceability matrices, can help identify the initial task list for the Product. It is important for training analysts to work with SMEs and human system interface experts to identify these tasks from a human perspective.

3.2.5 Review and validate the mission tasks

The final step in this process is to review and validate the identified mission critical tasks with SMEs and training program personnel. These SME's include system engineers, logistics specialists, human system interface personnel, and users. This review is performed during a user's conference.

3.3 Outputs

A documented list of missions (operational scenarios) of all specified, implied, and supporting missions for a Product, must be identified. A list of functions and critical tasks for each function should be included. The list of mission critical tasks should be recorded consistent with other task analysis activities including Product support analysis activities. Refer to in S3000L. The data elements associated with this analysis are described in Chap 3.7.

Chapter 3.3.2

Mission, job, and task analysis - Job analysis

Table of contents

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.3.1	Analysis - Mission, job, and task analysis - Mission analysis
Chap 3.7	Analysis - Business objects
S3000L	International procedure specification for Logistic Support Analysis (LSA)

1 General

A job is a collection of unique, specific, related activities that fulfill a specific role or function critical to achievement of an organization's mission or operational objective.

2 Description

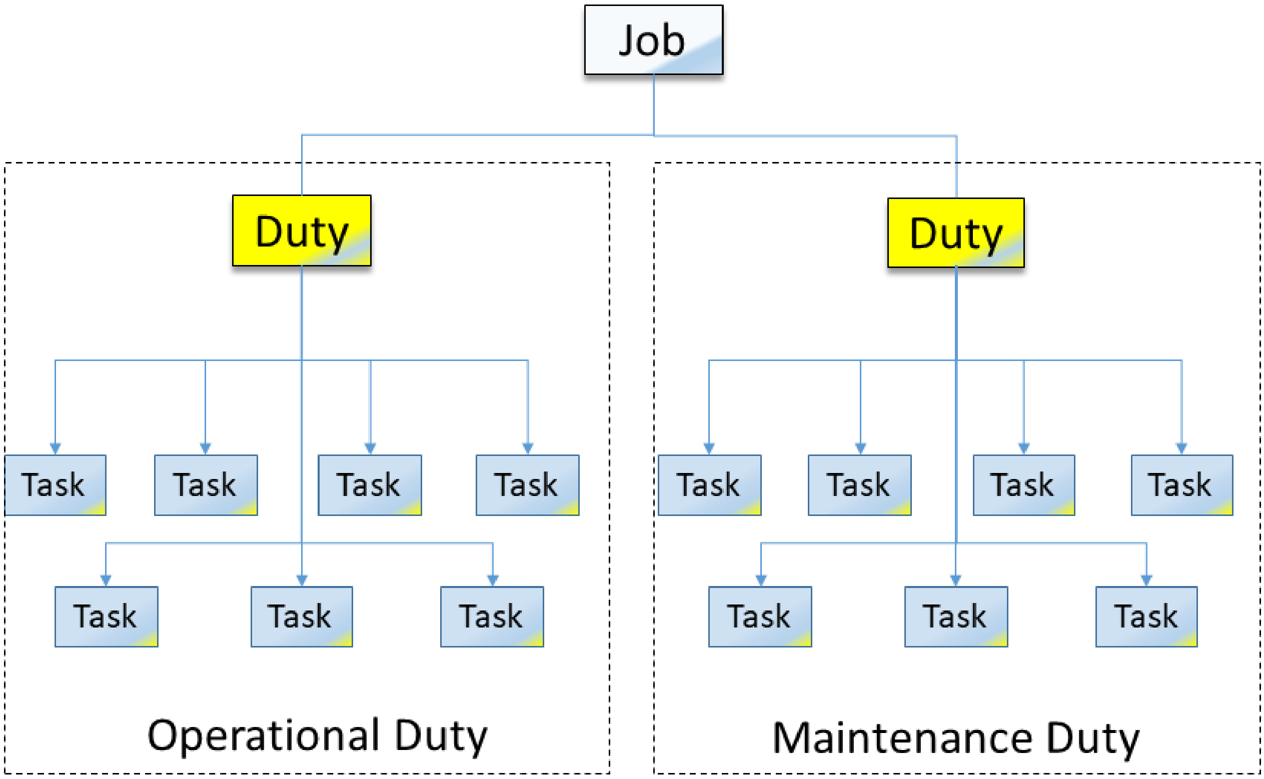
A job or role analysis is a systematic study to decompose, structure, and comprehensively describe a job for the purpose of establishing training requirements. It involves identifying all the individual critical tasks job holders perform to accomplish their missions and duties.

Whether developing a new training curriculum or updating an existing curriculum, data must be collected that will allow a job analysis to be conducted. Job analysis is a method used to obtain a detailed listing of tasks necessary to perform a specific job or duty. Job related data should include its purpose, functional responsibility of personnel, required support equipment and materials, and information on how the Product works, is maintained, or is used.

Collecting this data can involve observing personnel in the work environment, interviewing job incumbents and supervisors, questionnaire surveys, jury-of-experts, and the study of applicable occupational field descriptions, related training documents, and engineering data and specifications. The primary purpose for conducting a job analysis is to develop individual task lists.

Job analysis is undertaken to identify the human performance requirements of a Product or organization and to group these requirements into activities that are assigned to an individual. These human activities when organized into logical sets are called a job.

In a job analysis, a process is followed that breaks the job into duties and tasks, refer to Fig 1. A job is a group of major activities assigned to one individual. A job can be divided into functional units called duties. A duty is a distinct major division of work in a job. Each duty is composed of a group of related tasks within a given job (eg, bicycle repair). Duties have no specific start or finish and tend to be general in nature. Although identifying duties is not required, duty areas are a convenience in organizing tasks. A task represents a series of actions that leads to a meaningful outcome.

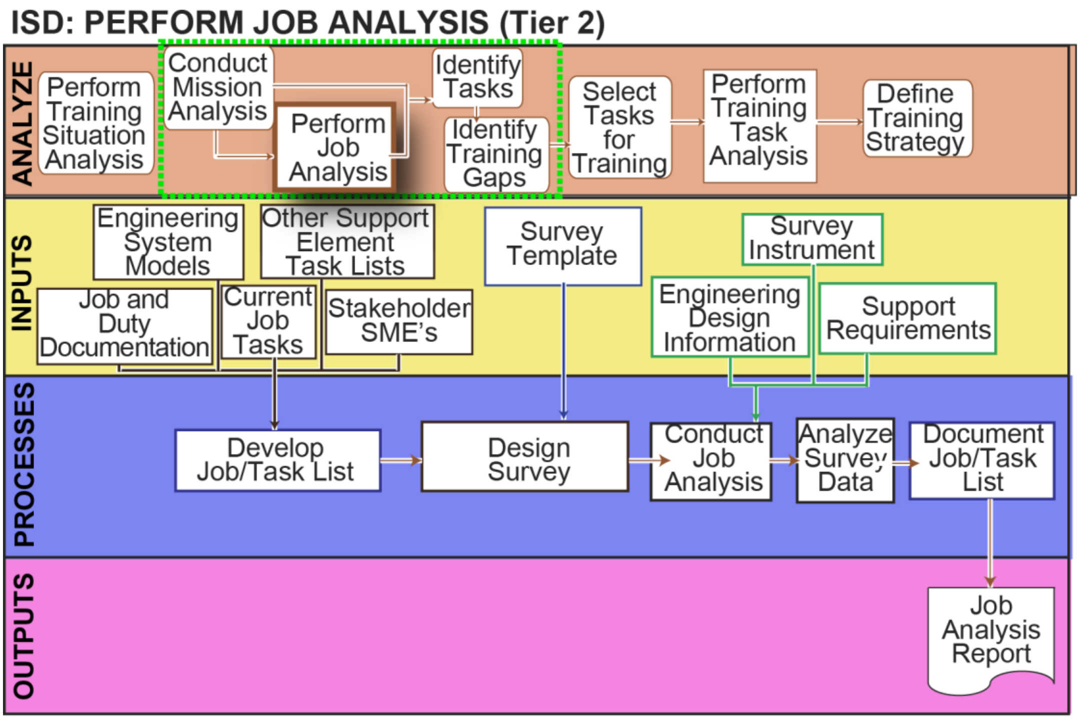


ICN-S6000T-B6865-00007-001-01

Fig 1 Job, duty, task breakdown

3 Process

The job analysis process is shown at Fig 2.



ICN-S6000T-B6865-00008-003-01

Fig 2 Perform job analysis

3.1 Inputs

The inputs for the job analysis process are:

- mission analysis, refer to Para 3.1.1
- job and duty documentation, refer to Para 3.1.2
- engineering product models, refer to Para 3.1.3
- current job/task, refer to Para 3.1.4
- other support element task lists, refer to Para 3.1.5
- stakeholder subject matter experts, refer to Para 3.1.6
- data collection sources, refer to Para 3.1.7
- support requirements, refer to Para 3.1.8

3.1.1 Mission tasks

The list of missions and mission critical tasks identified during mission analysis is a critical starting point for job analysis, refer to Chap 3.3.1. During this phase these mission critical tasks are grouped together by role.

3.1.2 Job and duty documentation

The identification of existing job and duty documentation including manpower appointments applicable to the Product are key aspects for performing a job analysis. These documents will provide information regarding additional duties and tasks that a job or role must perform in support of the Product.

3.1.3 Engineering Product models

Engineering Product models, including architectural and conceptual views, show the design of the Product. These views provide important human performance requirements and human interaction information as to how a Product will be used in support of mission.

3.1.4 Current job/task

Job/task information on similar jobs or roles, or for the same role on similar equipment, is important input data for a job analysis. Job analyses are performed for several reasons in addition to training development, so the results of similar job analysis should be considered when perform a job analysis for training purposes.

3.1.5 Other support element task list

Task lists from other support elements should be considered in performing a job/task analysis. These support elements include systems engineering, human factors engineering, and analyses detailed in S3000L.

3.1.6 Stakeholder subject matter experts

Subject Matter Experts (SME), who are performing the same or similar jobs on similar equipment, are a good source of information for determining operational or mission requirements and for detailing job and duty requirements of a Product.

3.1.7 Data collection sources

Data collection sources and techniques will be different when developing training for a new Product or developing training to accommodate a major modification to an existing Product, than when updating a training program to a mission or operational change. For example, in the early life of a new Product final technical documentation is not always available. Therefore, data collection sources and techniques will vary based on the situation. Typical data collection sources will include:

- interviews with engineers and SME
- surveys
- checklists
- occupational surveys
- small group panels
- similar job analysis reports

3.1.8 Support requirements

Support requirements developed by other Product support elements should also be considered. In addition to S3000L, other sources include:

- Integrated Support Plan (ISP)
- Product Support Requirements Specification (PSRS)
- Support System Design Description (SSDD)
- integrated master plan

3.2 Process steps

The job analysis process steps are:

- develop a job/task list, refer to Para 3.2.1
- determine the data collection techniques refer to Para 3.2.2
- conduct a job analysis, refer to Para 3.2.3
- analyze the data, refer to Para 3.2.4
- document the job/task list, refer to Para 3.2.5

3.2.1 Develop job/task list

The first step in the process is to gather and review all relevant data regarding job analysis and document an initial job/duty and task list. The input sources identified in Para 3.1 should be reviewed while creating the job/task list and a job (role), duty, task breakdown as shown in Fig 1 should be created. During this step the job should be examined to determine all of component duties and tasks, the conditions under which the job is performed, and the standards to be achieved when performing the job. The person in the job should also be considered. This will permit the identification of Knowledge, Skills, and Attitudes (KSA) necessary for effective performance. The job/duty/task breakdown should examine:

- job objective and responsibilities
- levels of supervision
- the conditions which the job or duty are performed
- job standards
- frequency of task performance
- consequences of inadequate task performance
- task criticality indication

3.2.2 Determine data collection techniques

Training analysts must plan their data collection requirements and techniques before beginning data collection. The training analyst should decide what data will be necessary for the project and should then develop a checklist for use during the data collection to ensure that all necessary data is collected. The data collection sheets should be discussed between all personnel that are carrying out the data collection to ensure that the data is collected consistently. More than one method should be used since each method has its inherent strengths and weaknesses.

Depending on the project requirements, the contents of the checklist include, but are not limited to:

- Product information:
    - how it works
    - how it is maintained
    - how it is used
- personnel functional responsibilities:
    - job Information:
        - job title
        - job purpose
        - duty title
        - job steps/procedures
        - location of performance
        - environment conditions
    - task information:
        - task titles
        - subtask titles
        - proper action verb assigned
        - supervision required
        - assistance required
        - physical demands
        - safety precautions
        - safety procedures

3.2.3 Conduct job analysis

Using the draft job/task lists and the appropriate data collection techniques the training analyst conducts the job analysis. The process for conducting the job analysis depends largely on the data collection methodology chosen, however the draft job/task list developed should be used as a starting point for each job/role.

The interview/survey method involves interviewing personnel currently serving in (or having recently served in) the job or role who have experience relevant to the subject area. Conduct this data collection method through face-to-face interviews and/or through a survey.

In the group interview, job incumbents are assembled to give information relative to their job. The training analyst asks questions about job performance and can ask the group to list data on tasks that cannot easily be demonstrated or observed. Because the group interview involves recall, rather than recognition, this method can provide inaccurate or incomplete data.

The observation/interview method involves sending the instructional analyst to observe end interview job incumbents and their supervisors on the job. Observing the job incumbents at work allows flexibility in gathering the required data by providing the training analyst opportunities to continually evaluate the information obtained. Direct observation of personnel as they perform their job, combined with interviews, provides the most useful source of task information. The training analyst should have a thorough understanding of the literature and functional relationships of the job to correctly interpret and describe the behaviors observed.

In the jury-of-experts method, experienced and knowledgeable personnel from various activities are brought together to record and analyze the data on jobs for which many critical behaviors are not directly apparent. This method can effectively supplement on-site observation/job analysis and written surveys. The experts are selected for their experience and knowledge of the job.

3.2.4 Analyze data

It is expected that the job analysis will produce a job/duty/task breakdown for each job that will operate or maintain a Product. The data collected from job incumbents in should be compared to the initial job/duty/task breakdown identified. A job should consist of a main objective of the role phrased in terms of the performance expected. Duties are the principal activities of the job holder and directly relatable to a job. They have no specific start or finish and are general in nature. Tasks are concise statements, beginning with a verb that specifically identifies major components of the role that result in a measurable outcome.

To ensure validity and assurance of the process, the job analysis activity should be governed by a dedicated steering group representing all stakeholders, as well as subject matter experts. This steering group will ensure that the data is analyzed correctly and completely, and all referenced data is available and reviewed. This steering group will vary in size and make-up based on the training requirement, the number of jobs to be analyzed, and the complexity of the Product.

3.2.5 Document job/task list.

The final step in this process is to document the job/task list for each job/role. This job/task list should include the information provided in Chap 3.7.

3.3 Outputs

A description of all identified jobs required to operate or maintain a Product is the critical output from this phase. The list of jobs should be recorded consistent with other job/task analysis activities including Product support analysis activities. Refer to S3000L. The business objects associated with this analysis are identified in Chap 3.7.

Chapter 3.3.3

Mission, job, and task analysis - Identify tasks

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.3.1	Mission, job, and task analysis – mission analysis
Chap 3.3.2	Mission, job, and task analysis - Job analysis
Chap 3.7	Analysis - Data element list
S3000L	International procedure specification for Logistic Support Analysis (LSA)
S4000P	International specification for developing and continuously improving preventive maintenance

1 General

To identify the tasks required for a particular job, it may be necessary for a training analyst to perform a task analysis. The goal of this process is to identify a complete task inventory for each job/role associated with the product. While this chapter describes the training process for performing a task analysis, it is recommended that training analysts work closely with other disciplines, including Logistics Support Analysis (LSA), to develop a comprehensive set of task inventories.

2 Description

The objective of the task identification is to develop task inventories that describe how humans will perform assigned Product functions and what results will be achieved by performing them. Human tasks include tasks that humans perform and that involve interaction with other parts of the Product. Human tasks are a critical foundation for the support of all Products. Task analysis is used to determine what the person must know and perform, specific requirements that must be met, and to establish minimum performance standards. This information is critical in performing a training needs analysis. Many disciplines employ some sort of Front-End Analysis (FEA) to determine task inventories.

Once task inventories have been identified, important task attributes or characteristics are defined, task interactions and sequences are identified, and an optimal task design is developed. Tasks are further defined as collective tasks, or individual tasks.

2.1 Tasks

2.1.1 Collective tasks

Collective tasks are tasks that require either or both:

- require more than one individual to complete, with person performing a discrete part of the task
- require simultaneous performance of task steps in different locations or requires many skills, that one person cannot perform the task in a timely or effective manner.

2.1.2 Individual tasks

Individual tasks are tasks that:

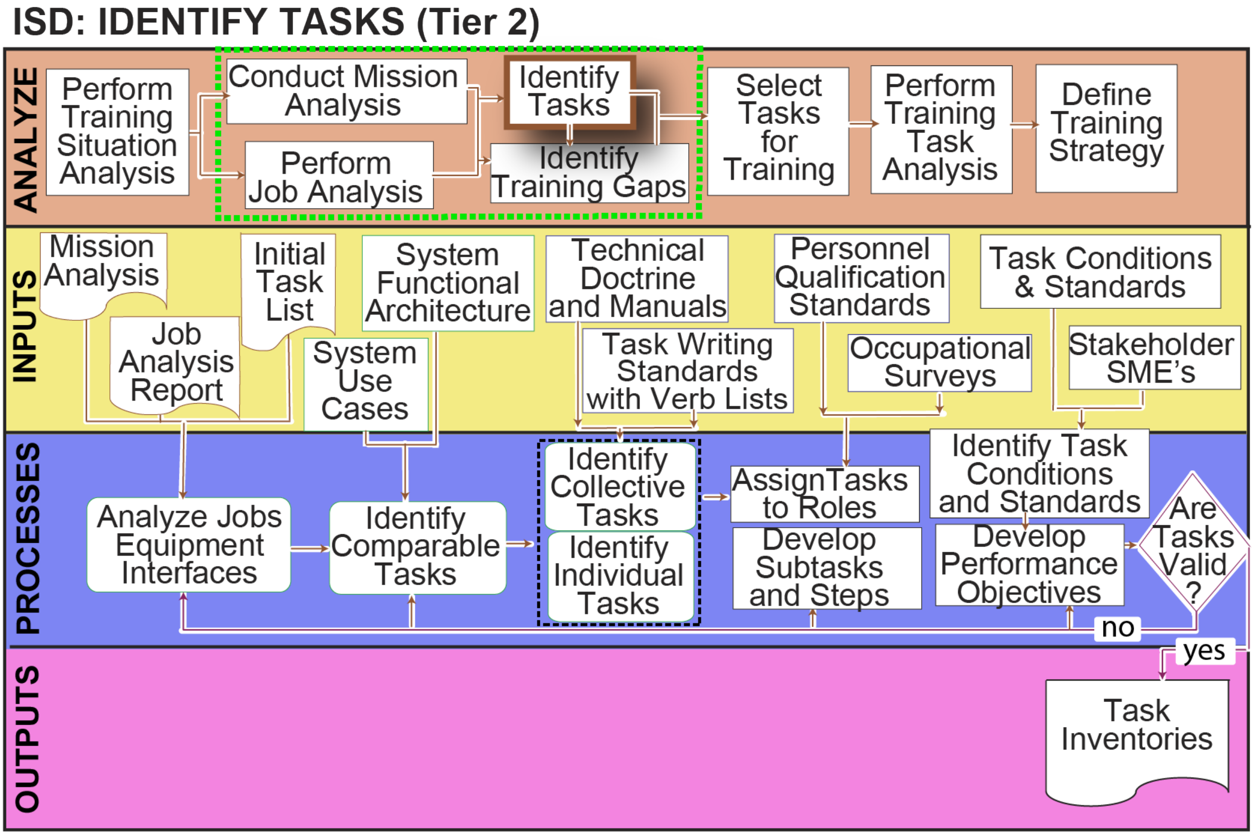
- have a definite beginning and end
- involve people interacting with or supporting the Product
- are directly observable or otherwise measurable
- are separate and distinct from other tasks
- include a mixture of decisions, perceptions, and/or physical activities required of a person

2.1.3 Task statements

As tasks are identified, a task statement is prepared that clearly define required behavior, conditions, and standards. A Performance Objective (PO) statement should be included that describes the entire intended action and shows that the action is specific and measurable. Task statements should be clear, complete, measurable, and relevant to the performance action. Refer to S3000L.

3 Process

The task identification process for individual tasks is shown in Fig 1.



ICN-S6000T-B6865-00009-003-01

Fig 1 Identify tasks process

3.1 Inputs

The inputs for the identify tasks process are the:

- initial task list, refer to Para 3.1.1
- system functional architecture, refer to Para 3.1.2
- Product use cases, refer to Para 3.1.3
- tactical doctrine and technical manuals, refer to Para 3.1.4
- personnel qualification standard, refer to Para 3.1.5
- occupational surveys, refer to Para 3.1.6
- task writing standards with verb list, refer to Para 3.1.7
- task conditions and standards, refer to Para 3.1.8
- stakeholder Subject Matter Experts (SME), refer to Para 3.1.9

3.1.1 Mission analysis

Mission analysis identifies what the Product is designed to accomplish (eg, personnel, transport, etc). It identifies the scenario/conditions, categories of factors or constraints under which the Product is expected to operate and be maintained (eg, time of day, all weather, all terrain operation, etc). Measures of performance, or performance standards, are analyzed to identify how well mission tasks must be performed to meet mission. Operational activities are analyzed to identify characteristics using operations templates, available requirements documents, Universal Joint Task List (UJTL) and in-service tactical level task lists, and inputs from SME.

3.1.2 Job analysis reports

Job analysis is a method used to obtain a detailed listing of tasks required to perform a specific job or duty. Job related data should include its purpose, functional responsibility of personnel, required support equipment and materials, and information on how the system works, is maintained, or is used. Collecting this data can involve observing personnel in the work environment, interviewing job incumbents and supervisors, questionnaire surveys, jury-of-experts, and the study of applicable occupational field descriptions, related training documents, and engineering data and specifications. The primary purpose for conducting a job analysis is to develop individual task lists. Refer to Chap 3.3.2.

3.1.3 Initial task list

Many programs perform a Logistical Support Analysis (LSA) and Human Systems Integration (HSI) analyses early in program development identifying maintenance and operational tasks, and sub-tasks. Refer to S3000L.

3.1.4 System functional architecture

Functional architecture identifies the full range of activities the Product is designed to perform. The functional architecture includes a function hierarchy that is produced using a top-down approach decomposing missions to, for example, flight phases, flight phases to functions, and functions to function steps.

3.1.5 Product use cases

Product use cases define the external objectives of Product performance. This information is extracted from system acquisition documentation (eg, capabilities design document, operational requirements document, Concept of Operations (CONOPS), performance-based specification, etc) that describe Product capabilities. Product capabilities are employed to satisfy user needs or to support mission achievement. Mission achievement is measured by effectiveness and is described in terms of the conditions and the standards to which each mission must be performed.

3.1.6 Tactical doctrine and technical manuals

Any tactical doctrines and technical manuals are a valuable source for identifying tasks. Technical doctrine included, but are not limited to, any concepts, principles, tactics, techniques, policies, procedures, etc, which guide the technical operations and support of a Product.

3.1.7 Personnel qualification standards

Personnel qualification standards is an evaluation system of basic performance designed to establish the minimum level of competency required for a member to successfully perform in their job.

3.1.8 Occupational surveys

Occupational surveys provide any job-related information regarding:

- physical demands
- environmental conditions
- education
- training and experience

- cognitive and mental requirements

3.1.9 Task writing standards with verb list

Task statements are prepared in a standard format to ensure intended action is clearly described and the action is specific and measurable. Task writing standards and verb lists help to ensure that task statements are clear, complete, concise, and relevant to the performance action.

3.1.10 Task conditions and standards

Task conditions are any special circumstances and environmental factors in which a task is to be performed. Task standards are the minimum acceptable proficiency required in the performance of the training task, under a specific set of conditions. Task conditions and standards are critical in determining the overall training strategy, assessment, and identifying specific training aides to be utilized.

3.1.11 Stakeholder subject matter experts

SME who are performing the same or similar jobs on similar equipment are a good source of information of task requirements.

3.2 Process steps

The identify tasks process steps are:

- analyze job equipment interfaces, refer to Para 3.2.1
- identify comparable tasks, refer to Para 3.2.2
- identify task lists, refer to Para 3.2.3
  - identify collective tasks, refer to Para 3.2.3.1
  - identify individual tasks, refer to Para 3.2.3.2
- assign tasks to roles, refer to Para 3.2.4
- develop subtasks and steps, refer to Para 3.2.5
- identify task conditions and standards, refer to Para 3.2.6
- develop task PO, refer to Para 3.2.7
- validate task lists, refer to Para 3.2.8

3.2.1 Analyze job equipment interfaces

The first step in the process is to analyze all inputs identified in Para 3.1 identifying any that meet the criteria for collective or individual tasks. This analysis should also identify any dependencies, including required tools and prerequisites. Successful task completion standards should also be identified. Tasks should be broken into their component behaviors or performances, representing actions, decisions, and paths as a sequence of behaviors.

3.2.2 Identify comparable tasks

Comparable tasks are tasks that have many of the same task steps or use similar or the same equipment performed in different physical locations of a Product. For example, calibrating the front bicycle brake and calibrating the rear bicycle brake. It is possible that comparable tasks only need to be analyzed once with reference to other tasks.

3.2.3 Identify tasks list

As tasks are identified, a task statement is prepared in a standard format. Using a standard format ensures that the intended action is clearly described, and the action is specific and measurable. The wording of the task statement consists of an action verb, an object, and possible qualifiers (optional). Task statements should be clear, complete, concise, and relevant to the performance action. Generally, when selecting action verbs for task statements, standardized lists of Knowledge, Skills and Attitude (KSA) verbs are used. Tasks should be clearly identified as either collective or individual. Refer to Para 2.1.1 and Para 2.1.2. The analysis should identify job roles, and skills required of the individuals so that a collective task can be performed and completed. Dependencies between job roles during collective task performance, as well as coordination requirements between job roles should also be determined. Collective tasks should be further analyzed to determine the individual tasks that make up the collective task. The intent of this step is to develop a complete set of tasks, with each task identified as either collective or individual.

3.2.3.1 Identify collective tasks

Collective tasks are tasks that require more than one person to complete, with each person performing a discrete part of the task or require simultaneous performance of task steps in different locations or contains such a large number of skills that one person cannot perform it in a timely or effective manner.

3.2.3.2 Identify individual tasks

An individual task is a task that requires one person to complete, with that person performing the entire task.

3.2.4 Assign tasks to roles

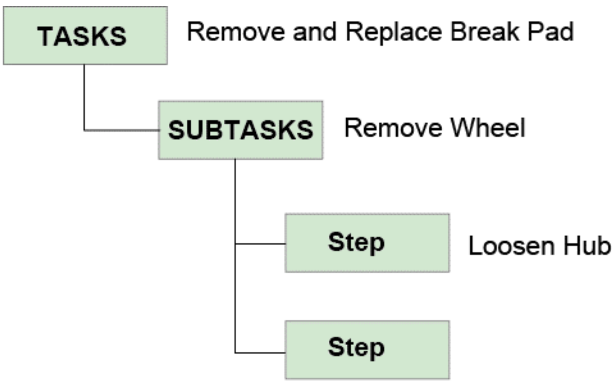
Once a complete set of tasks has been identified, each task should be assigned to the jobs or roles that are going to perform them so that a complete task inventory for each job or role can be created.

Note

A Task can be assigned to multiple jobs or roles.

3.2.5 Develop subtasks and steps

To determine the effective training requirements, tasks should be further decomposed into subtasks and steps. A subtask is collection of steps that comprise a major task activity, but are not done for their own sake, for example, Open aircraft panel. A step is a required unit/individual action that must be performed to accomplish the task/sub-task. Steps must be specific and detailed and contain only one action or unit of work, (eg, remove the bolt). Refer to Fig 2.



ICN-S6000T-B6865-00010-001-01

Fig 2 Task analysis decomposition

3.2.6 Identify task conditions and standards

Task conditions are circumstances and environmental factors in which the task is to be performed. Task standards are the minimum acceptable proficiency required in the performance of the task, under a specific set of conditions. These should be identified for every task that is to be trained. Task conditions and standards are likely to come from external sources such as S3000L for maintenance tasks, or human factors engineering for user tasks.

3.2.7 Develop task performance objective

A PO is a precise statement of the performance requirement expected in the execution of a task, expressed in terms of the conditions and standards to which it will be performed or demonstrated. A performance objective identifying appropriate task conditions and standards should be developed for each task and/or subtask.

3.2.8 Validate the task list

Task validation is conducted by a panel of SME who review each task statement, task conditions and task standards. All supporting information, including identified roles, skills, supporting equipment and task dependencies should be reviewed.

3.3 Outputs

The output is a task inventory for each job/role. Each task inventory will contain a set of PO describing each task's behavior, conditions, and standards. Task inventories should also include identified subtask and steps for each task.

Task information should be common across other disciplines and must not be interpreted as defined for S6000T alone. Preventive maintenance analysis should be recorded (refer to S4000P), and Product LSA (refer to S3000L), should be identified in accordance with these disciplines. The output is normally displayed in a task analysis worksheet with the required data elements. Refer to Chap 3.7.

Chapter 3.3.4

Mission, job, and task analysis - Training gaps

Table of contents

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.3.1	Mission, job, and task analysis - Mission analysis
Chap 3.3.2	Mission, job, and task analysis - Job Analysis
Chap 3.3.3	Mission, job, and task analysis - Task Analysis
Chap 3.4	Analysis - Task selection
Chap 3.6	Analysis - Training strategy
Chap 3.7	Analysis - Business objects

Chap No./Document No.	Title
Chap 4	Design
Chap 4.3	Design - Learning gaps
S3000L	International procedure specification for Logistic Support Analysis (LSA)

1 General

A training gap is defined as the gap between the behavior, conditions, and standards required to perform a new or changed task, and any pre-existing Learning Objectives included in a training curriculum.

A training gap occurs when there is a difference between a task's Performance Object (PO) and any pre-existing Learning Objectives (LOs) that are part of an existing curriculum.

The design of a new Product, or change to an existing product, is likely to:

- generate new tasks and new PO
- impact how current tasks are performed, thus affecting existing PO

These changes can result in training gaps, which are identified as the difference between the PO of a new or revised task and any pre-existing LO in an existing curriculum.

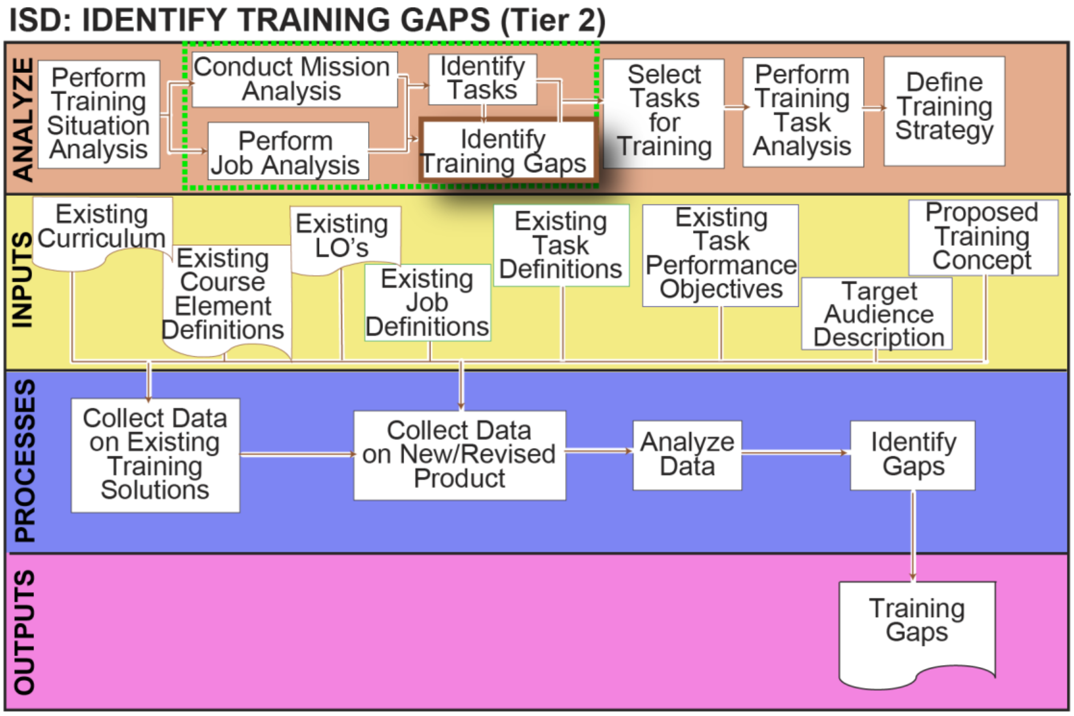
2 Description

The identification of training gaps involves analysis of current formal training solutions that address the KSA requirements to perform a job's tasks. These gaps are identified by comparing the KSA of new or affected tasks to existing LO statements, referenced by course elements in the existing training in the job's curriculum definition. The identification of training gaps is a determination of whether a training solution potentially exists for the new or changed task KSA requirements under analysis. If one does not exist, then the gap can be defined and tracked in the task selection analysis (Chap 3.4). If the task(s) with the KSA requirements are selected for training, then the identified training gap can continue to be tracked and later mapped to the training solutions defined during design processes (Chap 4). Additionally, training gaps that have tasks that are not selected for training can updated accordingly and provide a record of the decision to inform future analyses.

If it is determined that existing LO statements do not support training any aspect of a new or revised task's KSA requirements, then the magnitude of the training gap must be analyzed. This will depend on the nature of the Product change. If the task's KSA requirements are currently trained but affected by the Product change, then the training gap analysis should focus on the significant changes. For example, a bicycle design change resulted in manual caliper brakes being replaced by hydraulic disc brakes. If the KSA requirements for the task for replacing a tire are currently fully trained, then the gap analysis should focus on the change made to the braking system. If the affected task is partially trained or not trained at all, then the training gap is much more significant. Identifying the magnitude of training gaps can help optimize task selection and prioritization (Chap 3.4), the overall training strategy (Chap 3.6), as well as influence the design of new LO statements (Chap 4.3) used in training solutions that close the training gaps.

3 Process

Fig 1 shows the process for identifying training gaps.



ICN-B6865-00037-003-01

Fig 1 Identify training gaps

3.1 Inputs

The inputs for Identifying training gaps include:

- existing curriculum outlines, refer to Para 3.1.1
- existing course element definitions, refer to Para 3.1.2
- existing LO, refer to Para 3.1.3
- existing job definitions, refer to Para 3.1.4
- existing task definitions, refer to Para 3.1.5
- existing task PO, refer to Para 3.1.6
- target audience descriptions, refer to Para 3.1.7
- proposed training concept, refer to Para 3.1.8

3.1.1 Existing curriculum

If the Product being analyzed is a change to an existing Product, then current curriculum for the job being analyzed should be examined. If this is a new Product, then appropriate curricula for jobs defined for similar versions or variants of the Product should be examined. That examination includes course sequencing information, occupational definitions, total training time, as well as related Product context information.

3.1.2 Existing course element definitions

Course element information (eg, lessons, modules, sections) for existing course outlines for current or similar jobs should also be identified. Information including course element titles, description, LO statements addressed, training time, training methods and media used to support LO KSA requirements should be analyzed.

3.1.3 Existing learning objectives

Information on existing LO behaviors, conditions, and standards, as well as their level of learning and domain, should also be analyzed. Additional information based on Bloom's taxonomy, as well as LO relationship information should also be analyzed to understand how a LO is related to other LO (eg, an enabling LO that supports a terminal LO).

3.1.4 Job definitions

Information regarding the jobs defined to operate or maintain the Product should be identified. This information, including job/duty definitions and task lists, help define training requirements and establish the required training curriculum for the new or revised Product.

3.1.5 Task definitions

Task information for the Product being analyzed should be identified. Task information includes its name, description, resource requirements, and duration. Additional information about warnings and cautions associated with the tasks, as well as all subtask information should also be analyzed.

3.1.6 Task performance objectives

PO statements for the tasks being analyzed, including the expected behavior, conditions and standards in the job environment should be identified for all tasks associated with the Product and job being analyzed.

3.1.7 Target audience descriptions

The audience who will be trained to operate and maintain the Product should be identified and analyzed. This should include how trainees are selected for the training, anticipated attrition rates, current or anticipated class throughput, prior training, etc.

3.1.8 Proposed training concept

Training concepts for the new or revised Product should also be identified. This includes training methods, techniques, training technologies, as well as evaluation strategies.

3.2 Process steps

The process steps for identifying training gaps are:

- collect data on existing training solutions, refer to Para 3.2.1
- collect data on the new/revised Product, refer to Para 3.2.2
- analyze data, refer to Para 3.2.3
- identify training gaps, refer to Para 3.2.4

3.2.1 Collect data on existing solutions

The current environment includes anything that is similar or relates to, the new environment. This includes current training curricula, plans, and training material as well as current LO's. Information about the existing Product should also be collected. If the mission requirements, and jobs required to use and maintain an existing Product are similar to those of the new Product, then the existing information should also be collected.

3.2.2 Collect data on the new/revised Product

All available information for the Product being analyzed, including task information and PO, should be collected. For example, if the new Product is a replacement, a modification, or an upgrade to an existing Product, then this information on the new Product should be collected.

3.2.3 Analyze data

Data for the new or revised Product should be compared with the current training information. Questions to be examined include but are not limited to:

- Are the tasks for the new or revised Product currently trained, and are the training levels the same or have they changed?

- Have the PO for the new or revised Product changed significantly?
- Are the LO statements in the existing training appropriate for the new/revised tasks?
- How are the current LO statements trained and evaluated?
- Are they appropriate for the new/revised Product's requirements?

3.2.4 Identify training gaps

Identified training gaps should be clearly defined as part of a job's curriculum definition and associated to the task(s) that have caused them. A training gap definition should include, but is not limited to:

- an association to any changed or new task(s) that is not currently trained in existing training curricula
- an association to any task(s) that is partially trained, including information about what specifically is not being trained

3.3 Output

The output of this process is defined set of training gaps for a job's curriculum that are associated to the tasks included in a particular job definition.

Chapter 3.4

Analysis - Task selection

Table of contents

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.2	Analysis - Training situation analysis

Chap No./Document No.	Title
Chap 3.7	Analysis - Business objects
S3000L	International procedure specification for Logistic Support Analysis (LSA)
S4000P	International specification or developing and continuously improving preventive maintenance
DIF models	ADVISOR Enterprise Difficulty-Importance-Frequency (DIF) Model Fact Sheet
	Canadian Forces Individual Training & Education System 2003
JSP 822	Defence Systems Approach to Training - Direction and Guidance for Individual and Collective Training

1 General

Once the task inventories have been identified to the required level of detail and the tasks have been determined to be individual or collective, they are then analyzed to select those tasks that require training. It is often unnecessary or impractical to train every task that is required to perform a particular job. Financial constraints, time available and other constraints make it impractical to train every task.

2 Description

Training analysts categorize tasks based on criteria established to determine those that:

- require initial training
- can be performed using alternate methods, (eg, job aids)
- require no training

Identifying tasks that need to be trained is the first critical step in task selection, and this process should be followed carefully to ensure that the appropriate training level is identified for all tasks' KSA requirements. Wherever possible, appropriate information from the task inventory source(s) should be considered during this process.

2.1 Existing task selection models

Many critical task selection models are available for use in performing task selection analysis, however most of them are based on a similar model that examines a task's Difficulty, Importance and Frequency (DIF).

2.1.1 Florida State University model

This model is created by Florida State University and published in military regulations and standards (United States Department of Defense 2001) over the past 30 years. Based on a task's DIF, the model indicates whether the task should be trained, over trained, or not trained.

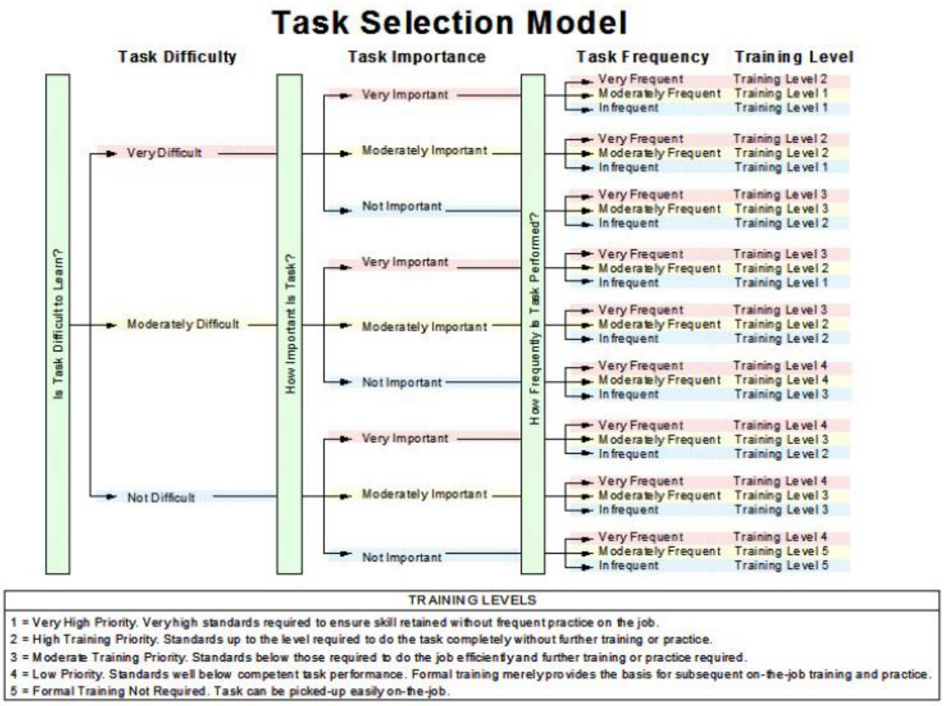
2.1.2 Criterion model

Published in the Canadian Forces Individual Training & Education System 2003. When making recommendations, in addition to a task's DIF, this model considers:

- the requirement for immediate action
- the percentage of individuals performing the task
- the consequences of inadequate performance

2.1.3 Ohio State Systematic Curriculum and Instructional Development DIF task model

The Systematic Curriculum and Instructional Development (SCID) model is similar to the Florida State DIF model but classifies training priorities into five levels ranging from a very high training priority (Level 1) thru to formal training not required (Level 5). These levels are based on the task's DIF using the algorithm. Refer to Fig 1.



ICN-S6000T-B6865-00011-001-01

Fig 1 Ohio State SCID algorithm

2.1.4 Joint Service Publication (JSP) 822

The United Kingdom Ministry of Defense direction and guidance for training education (refer to JSP 822) model aligns closely to the Ohio State SCID model and uses a similar model to identify six training categories (levels). Refer to Table 2.

Table 2 JSP 822 definitions

Training category	Definition
1	By the end of the formal training course the trainees will have performed the whole task several times, to the full job standard, and under realistic scenarios and conditions in which the operational physical, functional, and environmental fidelities were accurately reproduced. The trainee will be able to perform the task competently immediately on arrival in the operational workplace.
2	By the end of the formal training course the trainee will have performed the whole task at least once to full job standards, under realistic physical, functional, and environmental conditions and in a realistic scenario. The trainee should be able to perform the task on arrival in the operational workplace.
3	By the end of the formal training course the trainee will have performed the whole task in a training environment to a lesser standard than required in the job (ie, safety standards to be met in full).

Training category	Definition
4	By the end of the formal training course the trainee will have demonstrated an adequate level of underpinning knowledge and principles required but will not have applied it to develop the skills required to perform the task.
5	All formal training delivered in, or under the auspices of, the workplace.
6	Trainees do not require any formal training.

2.2 S6000T task selection model

The S6000T recommended selection model is based on the Ohio State SCID algorithm and JSP 822 algorithm and results in five training categories. Refer to Table 3.

Table 3 S6000T training levels and description

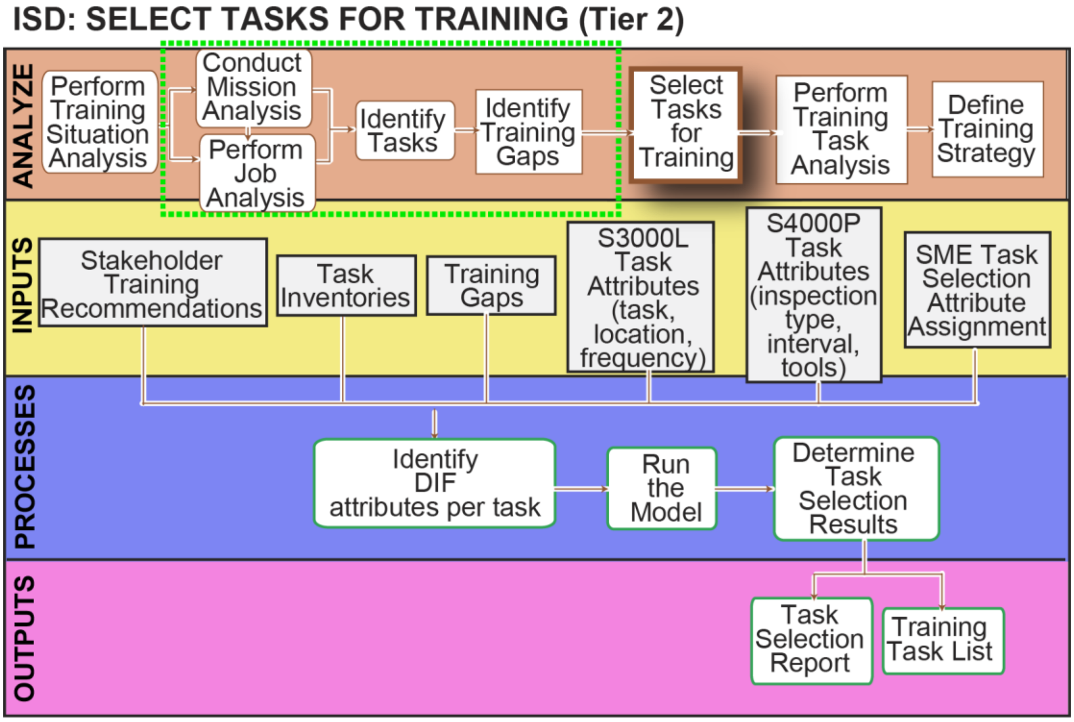
Level	Title	Description
1	Very high priority	Very high standards required to ensure skill retained without frequent practice on the job. By the end of the formal training course the trainees will have performed the whole task several times, to the full job standard, and under realistic scenarios and conditions in which the operational physical, functional, and environmental fidelities were accurately reproduced.
2	High training priority	Standards up to the level required to do the task completely. By the end of the formal training course the trainee will have performed the whole task at least once to full job standards, under realistic physical, functional, and environmental conditions and in a realistic scenario.
3	Moderate training priority	Standards below those required to do the job efficiently and further training or practice required. By the end of the formal training course the trainee will have performed the whole task in a training environment to a lesser standard than required in the job (ie, safety standards to be met in full).
4	Low priority	Formal training merely provides the basis for subsequent on-the-job training and practice. By the end of the formal training course the trainee will have demonstrated an adequate level of underpinning knowledge and principles required but will not have applied it to develop the skills required to perform the task
5	Formal training not required	Tasks can be picked-up easily on-the-job. Trainees do not require any formal training.

In addition to determining training levels and assisting in establishing training priorities, DIF analysis results should be fed back to design engineers. Tasks that are identified as difficult can be examined to determine whether automation, or some design changes can be made to the Product to make the tasks less difficult to perform. The results should also be fed back to the Product support analysis for additional support consideration or options.

3 Process

3.1 Select task for training

The task selection process starts with the stakeholder recommendations to identify a list of tasks that require training. Refer to Fig 2.



ICN-S6000T-B6865-00012-003-01

Fig 2 Task selection process

3.2 Inputs

The inputs to the task selection process are:

- stakeholder training recommendations, refer to Para 3.2.1
- task inventories, refer to Para 3.2.2
- training gaps, refer to Para 3.2.3
- S3000L task attributes, refer to Para 3.2.4
- S4000P attributes, refer to Para 3.2.5
- Subject Matter Expert (SME) task selection attribute assignments, refer to Para 3.2.6

3.2.1 Stakeholder training recommendations

In addition to task selection models, training recommendations of the stakeholder or customer must be considered. For example, some programs identify certain types of tasks (eg, remove and replace tasks) do not need to be trained. Other programs indicate that all tasks must be trained to some degree. In addition, stakeholders will help indicate if there are other performance support alternatives that can be considered instead of formal training. The training strategy of the stakeholder should also be examined. Some organizations try to minimize the amount of formal training, and stress on-the-job training. Other organizations have specific time allocated for formal training, which will impact the number of tasks that can be trained. Working with the stakeholder to determine these factors will help determine specific training strategy. Refer to Chap 3.2).

3.2.2 Task inventories

Task inventories are critical to get a complete picture of all the tasks that are required to operate or maintain the Product. These task inventories are a result of task analysis activities. There are multiple sources that perform these task analysis activities, including, but not limited to, product support analysis, program system engineering and human factor engineers. Training analysts should coordinate closely with these teams to ensure that a complete task inventory is compiled.

3.2.3 Training gaps

Identified training gaps (Chap 3.3.4) should be examined carefully when deciding training priorities. Tasks that generate large gaps in the KSA required for a job should especially be considered. The same task may generate very different training priorities based on the target audience performing them and the nature of the KSA that resulted in the related training gap.

3.2.4 S3000L task attributes

Several S3000L attributes should be considered when determining DIF levels. These attributes include, but are not limited to, task personnel safety criticality, warnings and cautions, task total labor time, task duration, task personnel resource, maintenance level, maintenance location, task Product integrity criticality, task operability impact, and task frequency. Training analysts should coordinate with Product support analysts to ensure these attributes are available.

3.2.5 S4000P attributes

Product support analysts normally identify task frequencies during their task analysis activities. For example, S4000P identifies the Preventative Maintenance Task Requirements with Intervals (PMTRI) that are used by S3000L in its Maintenance Task Analysis (MTA) and applied to S6000T. These task frequencies for preventive maintenance tasks determine how often a task needs to be performed and in conjunction with similar S3000L attributes are a critical attribute in performing a DIF analysis.

3.2.6 Subject matter experts task selection attribute assignments

Subject Matter Experts (SME) who are performing similar tasks or performing these new tasks should also provide input. They will provide input on how difficult the task is to train, as well as provide additional information not available from the other sources.

3.3 Process steps

The task selection process steps are:

- identify DIF attributes per task, refer to Para 3.3.1
- run the model, refer to Para 3.3.2
- determine task selection results, refer to Para 3.3.3

3.3.1 Identify DIF attributes per task

The training analyst should work with SME, as well as data from Product support analysis activities to answer the following questions regarding each task on the task list.

3.3.1.1 Difficulty

There are two possible difficulty attributes to measure:

- the difficulty of performing the task
- the difficulty of training the task

Different target organizations prefer different measures, however the S6000T DIF model uses difficult to perform. Difficulty is normally ranked as a low, medium, or high difficulty. A task that could cause harm or loss of life to a person if performed incorrectly should be considered difficult and would, for example, be identified as a highly difficult task.

3.3.1.2 Importance

In comparison to other tasks, it is essential to determine the importance of completing the task successfully in the operation of the Product (eg, if the task is not performed, will performance of the Product be degraded, or will it not work at all?). Depending on the task model utilized, this factor can be a binary (yes or no), or can be a ranking (eg, low, medium, or high). Tasks that could damage equipment if performed incorrectly should be considered important and highly important task.

3.3.1.3 Frequency

The frequency is a measure of how often a task normally performed. The scale can be in a fixed period (eg, how many times in a month, etc), or the scale can be during a typical operation of the Product (eg, how often this task is performed during each flight). This scale is normally very frequent, moderately frequent, or infrequent. Task groupings should be considered when determining task frequency. For example, if a task is to replace a flat tire on the bicycle, then the frequency should consider two tires on the bicycle.

3.3.2 Run the model

Each task should be run through the S6000T selection model described at Para 2.2. This model will recommend a training level of one through five to determine whether training is required and to what level.

3.3.3 Determine task selection results

The chosen selection model will result in a recommended training priority for each task in the inventory. The training analyst along with a SME should review the results and adjust accordingly, based on additional factors such as difficulty to train as well as other stakeholder training recommendations and guidelines. A final training priority level, as well as training level rationale should be indicated. Rationale should be identified for all training levels but must be identified for level six (ie, no formal training required). Examples of appropriate rationale include, but are not limited to:

- difficulty to train
- limited time
- automation eliminates need for training
- adequate help functions available
- low level of complexity
- excessive cost
- safety concerns
- use of hazardous material or environment required
- adequate performance support available
- security classification decision

4 Outputs

There are two outputs resulting from this process:

- task selection model results. The selection model results should clearly indicate the recommended training category for each task.
- tasks identified as requiring training. A complete list of all the tasks that require training should be identified. These tasks will continue through the rest of the analysis process.

5 Data model

Each task in the task inventory should have an attribute identified with it, indicating the final training decision associated for that task, as well as a rationale for tasks that do not require training. Refer to Chap 3.7.

Chapter 3.5

Analysis - Training task analysis

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.3.4	Mission, job, task analysis - Training gaps
Chap 3.4	Analysis - Task selection
Chap 3.7	Analysis - Data elements
Chap 4.4	Design - Learning objectives
ASD-STE100	Simplified Technical English
S3000L	International Procedure Specification for Logistic Support Analysis (LSA)

1 General

Once all the tasks that require training having been selected (refer to Chap 3.4), further analysis is conducted to define the learning level and Knowledge, Skills, and Attitudes (KSA) required for each. This is accomplished by analyzing the Performance Objective (PO) for each task/subtask, and then analyzing its attributes and training decision characteristics in relation to KSA taxonomies.

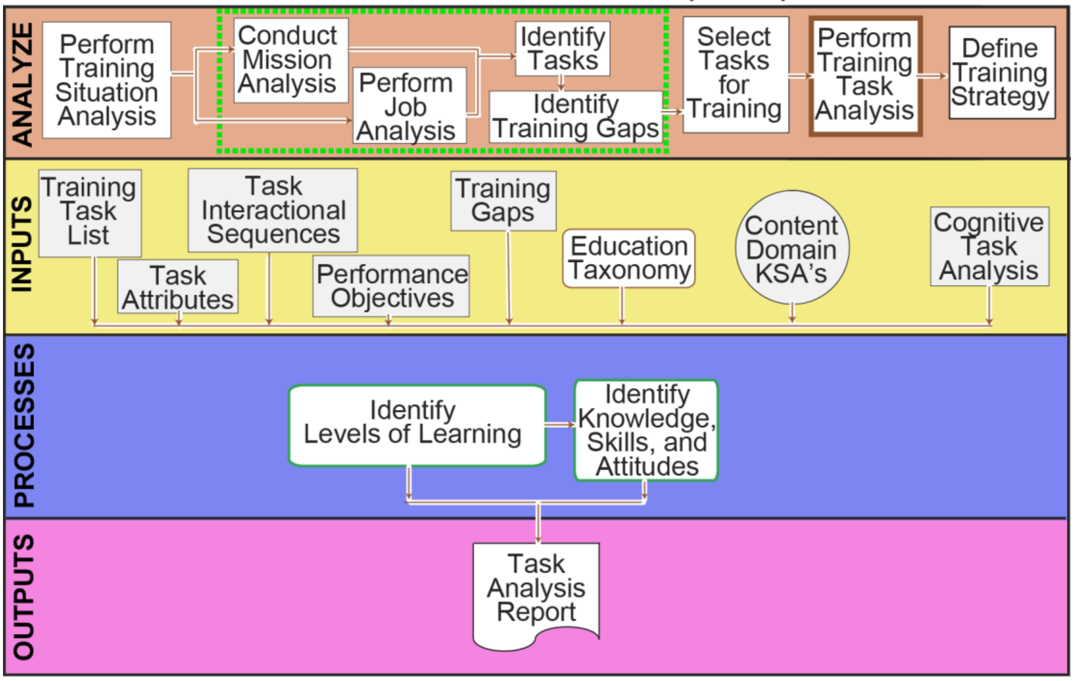
2 Description

The task performance details should describe how the task is performed, under what conditions it is performed and to what standard the task must be performed. The level of learning and KSA required to perform the tasks and subtasks, as well as additional task attribute data, help the training analyst establish the individual training strategy, and design and develop the training programs and products.

3 Process

The training task analysis process is shown at Fig 1.

ISD: PERFORM TRAINING TASK ANALYSIS (Tier 2)



ICN-S6000T-B6865-00013-003-01

Fig 1 Training task analysis - Tier 2

3.1 Inputs

The inputs for the training task analysis process are:

- training task list, refer to Para 3.1.1
- task attributes, refer to Para 3.1.2
- task interactional sequences, refer to Para 3.1.3
- PO, refer to Para 3.1.4
- training gaps, refer to Para 3.1.5
- education taxonomy, refer to Para 3.1.6
- content domain KSA, refer to Para 3.1.7
- cognitive task analysis, refer to Para 3.1.8

3.1.1 Training task list

The training task list contains all the tasks that are identified as requiring some level of formal training from the task selection process (refer to Chap 3.4). All identified tasks/subtasks that are labelled as requiring additional training analysis.

3.1.2 Task attributes

Task attributes are additional information regarding a task, including task frequency, task difficulty, task resources, type of task, etc. These attributes help the training analyst identify the overall training strategy for the task, as well as identifying additional training resources requirements. These attributes often come from other sources, such as human factors engineering, as well as the Operational Task Analysis (OTA) and Maintenance Task Analysis (MTA), (refer to S3000L), activities.

3.1.3 Task interactional sequences

Task interactional sequences are any dependencies between tasks. This could be where certain tasks must be performed before other tasks can be started (eg, an Investigate bird strike task can result in a Shutdown engine in flight task) or where certain tasks are dependent on the resulting data from another task (eg, a task to determine a tire pressure value can result in a task to inflate/deflate the tire pressure).

3.1.4 Performance objectives

The PO associated with each task provides the task's behavior, the conditions, which are any circumstances and environmental factors in which a task will be performed, as well the task standards, which are the minimum acceptable proficiency levels required for the task performance. Each of these factors play an important role in determining the required level of learning for each task, as well as the KSA requirements associated with each task.

3.1.5 Training gaps

Training gaps (refer to Chap 3.3.4) provide an initial analysis of a task's KSA requirements of a task. They identify KSA requirements that are not currently addressed in any current or related curriculum where the same or similar tasks are trained.

3.1.6 Education taxonomy

KSA taxonomies are used to define the verbs used in each learning objective. Appropriate cognitive (knowledge), affective (attitude) and psychomotor (skill) taxonomies should be chosen prior to beginning the training task analysis process.

Note

There are many taxonomies available, but the most used, and recommended, are Bloom's taxonomy and EJ Simpson's taxonomy (based on the work of Benjamin Bloom).

3.1.6.1 Cognitive taxonomy

The cognitive domain relates to knowledge and includes the recall or recognition of specific facts, patterns and concepts that serve in, and the development of intellectual abilities. In Bloom's taxonomy there are six major categories. Refer to Table 2.

Table 2 Bloom's cognitive (knowledge) taxonomy

Category	Definition
Knowledge/ Remembering	Recalling or retrieving previously learned data and information (facts, rules, concepts, and principles).
Comprehending/ Understanding	Comprehending the meaning, translation, interpolation, and interpretation of instructions and problems. State a problem in one's own words.
Applying	Attaining and applying knowledge. Understanding data and information (facts, rules, concepts, and principles) in the context of work.
Analyzing	Separating material or concepts into component parts so that their organizational structure can be understood. Distinguishing between facts and inferences.
Evaluating	Making a judgment about the value of ideas or materials. It is the evaluation of reasoning, using specific criteria to evaluate and then decide. Using a set of criteria and responding effectively to unexpected experiences, present and defend opinions by making judgments about information, the validity of ideas and the quality of work.

Category	Definition
Synthesize	Building a structure or pattern from diverse elements. Putting parts together to form a whole, with emphasis on creating a new meaning or structure.

3.1.6.2 Affective taxonomy

The affective domain relates to the way individuals deal with things emotionally, such as feelings, values, appreciation, enthusiasm, motivations, and attitudes. In Bloom's taxonomy there are five major categories. Refer to Table 3.

Table 3 Bloom's affective (attitude) taxonomy

Category	Definition
Receiving	The ability to perceive the normal, abnormal, and emergency cues associated with the performance of an operational procedure. Situational awareness of operational cues.
Responding	The mental preparedness to encode operational cues as indicators of normal, abnormal, and emergency conditions associated with the performance of an operational procedure.
Valuing	The ability to judge the worth or quality of normal, abnormal, and emergency cues associated with the performance of an operational procedure.
Organizing	The mental preparedness to make decisions, by using prioritized strategies and tactics in response to normal, abnormal, and emergency condition cues associated with the performance of an operational procedure.
Internalizing	The mental preparedness to make decisions by generating the results expected upon completion of prioritized strategies or tactics. It includes the response to normal, abnormal, and emergency cues associated with the performance of an operational procedure, and the ability to generate new prioritized strategies and tactics.

3.1.6.3 Psychomotor taxonomy

The psychomotor domain relates to the development of physical movement, coordination and skills which are normally achieved through repetitive practice (eg, the ability to manipulate a tool or piece of equipment). E.J. Simpson's taxonomy, there are eight major categories. Refer to Table 4.

Table 4 Simpson's psychomotor domain

Category	Definition
Perception (awareness)	The ability to use sensory cues to guide motor activity. This ranges from sensory stimulation, through cue selection, to translation.
Set	The readiness to act, to include mental, physical, and emotional sets. These three sets are dispositions that predetermine a

Category	Definition
	person's response to different situations (also known as mindsets).
Guided Response	The early stages in learning a complex skill that includes imitation and trial and error. Adequacy of performance is achieved by practicing.
Mechanism (basic proficiency)	The intermediate stage in learning a complex skill. Learned responses have become habitual and the movements can be performed with some confidence and proficiency.
Complex overt response (expert)	The skillful performance of motor acts that involve complex movement patterns. Proficiency is indicated by a quick, accurate and highly coordinated performance, requiring a minimum of energy. This category includes performing without hesitation and automatic performance.
Adaptation	This category includes skills that are well developed, and the individual can modify movement patterns to fit special requirements.
Origination	Creating new movement patterns to fit a particular situation or specific problem. Learning outcomes emphasize creativity based upon highly developed skills.

3.1.7 Content domain KSA

For maintenance tasks, the MTA activities can provide information on personnel resources required to conduct a task and any skill information, including skill codes, required to support a task, (refer to S3000L). Similar attributes should be available from the OTA.

3.1.8 Cognitive task analysis

Cognitive task analysis is a type of task analysis aimed at understanding tasks that require cognitive activity from the user, such as decision-making, problem-solving, memory, attention, and judgment.

3.2 Process steps

The training task analysis process steps are:

- identify learning level, refer to Para 3.2.1
- identify required KSA, refer to Para 3.2.2

3.2.1 Identify learning level

The PO for each task and subtask should be analyzed against the cognitive, psychomotor, and affective taxonomies, and categorized based on its KSA requirements. The conditions and standards of the PO should be considered in making these decisions. Each task and subtask must be given a learning level within the taxonomies (eg, complex overt response).

3.2.2 Identify required KSA

Each task's/subtask's PO, plus the appropriate Bloom's taxonomy, should be used to determine the relevant KSA required to perform each task/subtask. KSA can be assigned at all levels, including tasks, subtasks, and steps. Lower learning level KSA are generally associated with subtasks and steps, whereas higher learning level KSA are associated at the task level.

A verb for the KSA statement should then be chosen from the identified learning level (eg, fix). Table 5 thru Table 20, and guidance on verbs provided by simplified technical English, should be used when categorizing, assigning learning levels, and selecting KSA verbs. Refer to ASD-STE100. Verbs contained within the relevant taxonomies should be used to facilitate the development of Learning Objectives (LO) (refer to Chap 4.4) during the design phase.

Table 5 Knowledge (cognitive) – Remember

Remember verbs		
Answer	Arrange	Apply
Calculate	Categorize	Collect
Coach	Compare	Complete
Confirm	Contrast	Describe
Designate	Differentiate	Discriminate
Distinguish	Divide	Duplicate
Elaborate	Eliminate	Exchange
Express	Extract	Finalize
Find	Follow Directions	Gather
Group	Identify	Instruct
Label	List	Locate
Match	Memorize	Notify
Order	Organize	Outline
Quote	Rank	Read
Realign	Rebuild	Recite
Recognize	Recommend	Record
Recreate	Repeat	Reproduce
Respond	Restate	Schedule
Select	Tell	Train
Tune		

Table 6 Knowledge (cognitive) - Comprehension

Comprehension verbs		
Appraise	Change	Classify
Compile	Compose	Correct
Define	Depict	Describe
Discriminate	Discuss	Estimate
Explain	Express	Extend
Find (locate)	Find the difference	Format

Comprehension verbs		
Forward	Give example	Identify
Illustrate	Indicate	Infer
Locate	Measure	Obtain
Outline	Paraphrase	Predict
Put in order	Recognize	Record
Report	Restate	Review
Rewrite	Select	Simplify
Subtract	Summarize	Transcribe
Translate		

Table 7 Knowledge (cognitive) - Application

Application verbs		
Advise	Allocate	Assume
Call	Clear	Condense
Convert	Deliver	Demonstrate
Determine	Develop	Differentiate
Discover	Discuss	Dispense
Distinguish	Draw	Edit
End	Estimate	Execute
Facilitate	Finish	Gauge
Host	Illustrate	Implement
Interpret	Investigate	Keep records
Locate	Log	Manipulate
Map	Modify	Organize
Participate	Pause	Perform
Persuade	Plan	Practice
Predict	Present	Prioritize
Process	Produce	Proof
Prove	Provide	Reclaim
Relate	Resume	Retrieve
Scan	Set up	Ship
Show	Sign on	Situate

Application verbs		
Sketch	Solve	Store
Submit	Supply	Terminate
Trace	Transfer	Translate
Train	Teach	

Table 8 Knowledge (cognitive) - Analysis

Analysis verbs		
Analyze	Arrange	Categorize
Challenge	Classify	Contrast
Criticize	Debate	Deduce
Detect	Derive	Determine
Diagnose	Diagram	Differentiate
Discriminate	Experiment	Formulate
Generalize	Illustrate	Infer
Inspect	Interview	Organize
Outline	Probe	Process
Question	Relate	Select
Separate	Solve	Survey
Test	Troubleshoot	Verify

Table 9 Attitude (affective) - Receiving

Receiving verbs		
Acknowledge	Ask	Attend
Be aware	Choose	Describe
Directing attention	Follow	Getting attention
Give	Hold	Holding attention
Locate	Reply	Show alertness
Tolerate	View	Watch

Table 10 Attitude (affective) - Responding

Responding verbs		
Accomplish	Achieve	Acknowledge
Advise	Aid	Allow

Responding verbs		
Announce	Answer	Anticipate
Apologize	Ask	Assist
Communicate	Complete	Comply
Conform	Consent	Contact
Contribute	Cooperate	Demonstrate
Describe	Discipline	Discuss
Follow-up	Greet	Indicate
Interpret	Label	Notify
Participate	Question	Permit
Praise	Pursue	React
Refuse	Reply	Request
Respond	Resume	Seek
Visit	Volunteer	Welcome
Entertain	Enquire	

Table 11 Attitude (affective) - Valuing

Valuing verbs		
Adopt	Appreciate	Choose
Commit	Contemplate	Desire
Differentiate	Display	Dispute
Endorse	Enjoy	Ensure
Exhibit	Explain	Express
Form	Initiate	Invite
Judge	Justify	Participate
Prefer	Propose	Report
Sanction	Share	Study

Table 12 Attitude (affective) - Organize

Organize verbs		
Adhere	Alter	Approve
Arrange	Care	Categorize
Characterize	Classify	Combine
Coordinate	Defend	Establish

Organize verbs		
Explain	Formulate	Generalize
Group	Integrate	Modify
Observe	Order	Perceive
Rank	Rate	Recognize
Reconnoiter	Relate	Synthesize
Systemize	Organize	Perceive
Rank	Rate	Recognize
Reconnoiter	Relate	Synthesize
Systemize		

Table 13 Attitude (affective) - Characterizing

Characterizing verbs		
Act	Advocate	Alert
Alter	Approve	Assess
Assume	Authenticate	Behave
Believe	Characterize	Choose
Command	Conceive	Conform
Conjecture	Conserve	Constitute
Continue	Coordinate	Defend
Develop	Devise	Devote
Disclose	Discriminate	Display
Encourage	Endure	Enforce
Ensure	Exemplify	Exonerate
Favor	Formulate	Function
Imagine	Incorporate	Influence
Innovate	Judge	Justify
Maintain	Modify	Perform
Practice	Prescribe	Preserve
Prioritize	Promote	Propose
Qualify	Question	Rally
Rationalize	Reassess	Reserve
Respect	Retain	Review
Revise	Sell	Serve

Characterizing verbs		
Share	Support	Study
Uphold	Use	Validate
Verify	Vindicate	

Table 14 Skill (psychomotor) - Perception (awareness)

Perception (awareness) verbs		
Choose	Describe	Detect
Differentiate	Distinguish	Isolate
Relates	Attempt	Delineate
Detect	Diagram	Follow
Draw	Enlarge	Fall
Feel	Hear	Jump
Listen	look	Organize
Pierce	Sketch	Scan
See	Separate	Smell
Stay	Strengthen	Stretch
Taste	Throw	Touch
Try	Twist	Visualize
Wear		

Table 15 Skill (psychomotor) - Set

Set verbs		
Begin	Carry	Creep
Depart	Define	Display
Explain	Outline	Perform
Proceed	React	Respond
Shift	State	Run
Volunteer		

Table 16 Skill (psychomotor) - Guided response

Guided response verbs		
Begin	Blueprint	Break up
Construct	Copy	Depict

Guided response verbs		
Dissect	Disturb	Discompose
Disjoin	Divide	Draft
Duplicate	Follow	Imitate
Mimic	Mock	Practice
React	Reinforce	Replicate
Represent	Reproduce	Respond
Scatter	Shorten	Split
Trace	Follow	Respond
React		

Table 17 Skill (psychomotor) - Mechanism (basic proficiency)

Mechanism (basic proficiency) verbs		
Acquire	Advance	Assault
Break up	Carry	Conduct
Dismantle	Disperse	Display
Distribute	Do	Drive
Execute	Fasten	Fix
Follow	Guide	Handle
Hover	Improve	Insert
Join	Land	Load
Maintain	Maneuver	Manipulate
Pace	Perform	Produce
Progress	Raise	Regulate
Sketch	Steer	Stoop
Take off	Track	Transfer
Transport	Traverse	Use
Wire		

Table 18 Skill (psychomotor) - Complex overt response (expert)

Complex overt response (expert) verbs		
Acquire	Advance	Assault
Break up	Carry	Conduct
Dismantle	Disperse	Display

Complex overt response (expert) verbs		
Distribute	Do	Drive
Execute	Fasten	Fit (parts together)
Fix	Follow	Guide
Handle	Hover	Improve
Insert	Join	Land
Load	Maintain	Maneuver
Manipulate	Pace	Perform
Produce	Progress	Raise
Regulate	Sketch	Steer
Stoop	Take off	Track
Transfer	Transport	Traverse
Use	Wire	

Note

The complex overt response (expert) verbs are the same as the mechanism verbs, but will have adverbs or adjectives that indicate that the performance is quicker, better, more accurate, etc.

Table 19 Skill (psychomotor) - Adaptation

Adaptation verbs		
Acclimatize	Accommodate	Adjust
Alter	Ambush	Attack
Bypass	Conduct	Deploy
Direct	Draw	Evade
Fit	Fix	Flip
Grasp	Infiltrate	Invert
Lay	Lead	Map
Modify	Modulate	Mutate
Navigate	Neutralize	Occupy
Orient	Oscillate	Pack
Patrol	Prevent	Program
Protect	Queue	Readjust
Rearrange	Reconcile	Reconstitute
Reconstruct	Recover	Reduce
Regulate	Relieve	Reorder

Adaptation verbs		
Reorganize	Reorient	Reshuffle
Retool	Revert	Revise
Rig	Spin	Square
Suppress	Surpass	Swing
Swirl	Tailor	Temper
Train	Transcend	Transpose
Twirl	Vary	

Table 20 Skill (psychomotor) - Origination

Origination verbs		
Acquire	Arrange	Assemble
Blend	Break up	Build
Carry out	Cause	Combine
Complement	Complete	Compose
Conduct	Connect	Constitute
Construct	Contrive	Create
Design	Dismantle	Disperse
Distribute	Drive	Enact
Encircle	Enclose	Establish
Evolve	Execute	Handle
Improve	Incorporate	Initiate
Insert	Invent	Involve
Link	Load	Manipulate
Merge	Originate	Perform
Produce	Progress	Raise
Refine	Rewire	Stoop
Transcend	Transfer	Transform
Transport	Unite	

3.3 Outputs

3.3.1 Task analysis report

The outputs for this process include identified learning levels for the KSA requirements of each task and subtasks identified for formal training. and knowledge, skill, and attitude statements. Refer to Chap 3.7.

Chapter 3.6

Analysis - Training strategy

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.2	Analysis - Training situation analysis
S3000L	International procedure specification for Logistic Support Analysis
S4000P	International specification for developing and continuously improving preventive maintenance

1 General

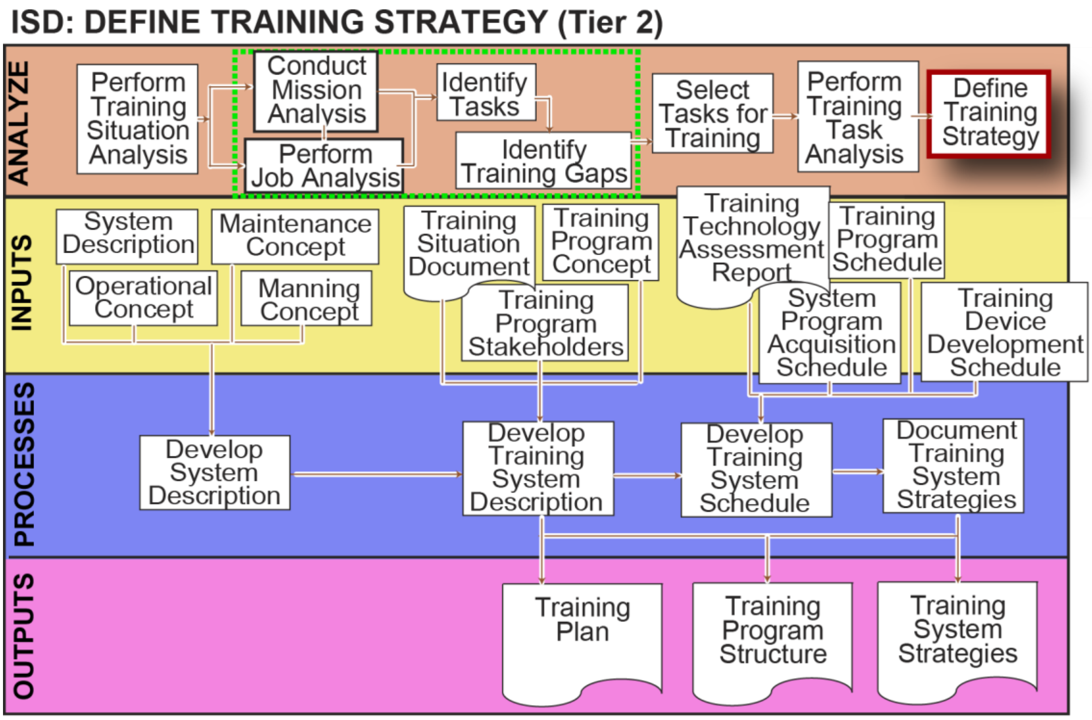
Once the Knowledge, Skills, and Attitudes (KSA) have been identified for the tasks that require training the next step is to identify the strategy for achieving them.

2 Description

A training strategy is a blueprint for training development to meet the training needs of the organization and support mission accomplishment. It must be aligned to the overall strategy of the Product so that the Product can be operated and maintained upon delivery.

3 Process

There are many inputs available to help in defining an effective training strategy. Refer to Fig 1.



ICN-S6000T-B6865-00014-001-01

Fig 1 Define training strategy

3.1 Inputs

An effective training strategy requires:

- a system description, refer to Para 3.1.1
- an operational maintenance concept, refer to Para 3.1.2
- a training situation document, refer to Para 3.1.3
- a training program concept, refer to Para 3.1.4
- a list of training program stakeholders, refer to Para 3.1.5
- a training technology assessment report, refer to Para 3.1.6
- a system program acquisition schedule, refer to Para 3.1.7

3.1.1 System description

It is essential that the training strategy supports the Product. A description of the Product including system requirements, operating environment, system and subsystem architecture, human machine interfaces as well as detailed design should be used in developing an effective training strategy.

3.1.2 Operational and maintenance concept

Operational and maintenance concept is a critical input for developing an effective training strategy. Operational concepts including environmental conditions in which the Product will operate, as well as manpower operational requirements should be included. Maintenance concepts, including maintenance levels, and type of maintenance actions to be performed for the overall Product as well as individual systems should also be included.

3.1.3 Training situation document

The training situation document, which is the result of the Training Situation Analysis (TSA) (refer to Chap 3.2) is also a critical input for this phase. The training situation document records the gaps in the current or similar Products and identifies how the current Product will help fill these gaps.

3.1.4 Training program concept

The training program concept is the conceptual plan for the training. This includes:

- identifying how much formal training will be available
- a decision on whether the training be done at a formal training location, or performed at operational sights, etc
- a decision on what is the planned level of training the students will be at the conclusion of training will they be experts, or merely exposed to key concepts, with practice occurring on the job

This gap should be clearly documented and agreed to by all senior leadership.

3.1.5 Training program stakeholders

The input of the stakeholders of the planned training program are critical for determining the appropriate training strategy. Stakeholders include program management, senior training leadership, current training subject matter experts as well as the intended training audience.

3.1.6 Training technology assessment report

The training technology assessment is part of the training situation document detailed as part of the training situation analysis (refer to Chap 3.2). The training technology assessment data contains information on technology used to provide training for existing systems, which are similar to the emerging system for which training must be developed. It also provides an assessment of the training technologies that are used in areas relevant to the training requirements of the emerging system.

3.1.7 System program acquisition schedule

The system program acquisition schedule is part of the Program's integrated master schedule and details the delivery schedule for all portions of the Product including the support system. The acquisition schedule should also indicate when the training system needs to be complete and ready for implementation.

3.2 Process steps

The process steps for defining a training strategy are:

- develop system description, refer to Para 3.2.1
- develop training system description, refer to Para 3.2.2
- develop training system schedule, refer to Para 3.2.3
- document training system strategies, refer to Para 3.2.4

3.2.1 Develop system description

The system description provides an overview of the Product and describes the system requirement, operating and maintenance environments of the Product, the system and subsystem architecture, and a detailed design description.

3.2.2 Develop training system description

The training system description provides an overview of the complete training system used to support the Product and documents the detailed training requirements. It should include but limited to:

- purpose the training system
- goals and constraints of the training
- number of Category of Skill levels requiring training
- identification and description of training equipment
- facility requirements
- personnel resources required to develop, implement, and maintain the training system

3.2.3 Determine training system schedule

A detailed schedule should be identified for the entire training program. It should identify when any formal training will begin, as well as the acquisition schedule for training assets such as training equipment, curriculum, as well as support equipment.

3.2.4 Document training system strategy

The training strategy should be clearly documented. The training strategy should:

- define the goals and constraints for training
- list the location(s) of instruction
- list the anticipated student load
- describe the training approach based on the tasks to be trained and the availability of appropriate training media
- the identified training schedule including plans of action and milestones
- describe personnel resources required to develop, implement and maintain the training system
- list all training facility requirements
- describe how the training will be evaluated and how the student performance will be assessed
- describe quality management to include process and Product compliance with approved plans, procedures, and processes
- identify the funding requirements, including Life Cycle Costs (LCC), total ownerships costs and whole life costs of the training program. Refer to S3000L.

A training system is a critical element of Product support. As such processes for Logistics Support Analysis, as well as improving preventative maintenance apply to the training system as well as the overall Product being supported. Refer to S4000P.

3.3 Outputs

3.3.1 Detailed training strategy

The output of this phase is a formal training strategy, including a management plan with detailed schedule and milestones, as well as identification of the appropriate strategies that will be used to provide the trainees with the required KSA.

Chapter 3.7

Analysis - Business objects

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 3.1	Analysis - General
Chap 3.2	Analysis - Training situation analysis
Chap 3.3	Analysis - Mission, job, and task analysis
Chap 3.3.1	Mission, job, task analysis - Mission analysis
Chap 3.3.2	Mission, job, task analysis - Job analysis
Chap 3.3.3	Mission, job, task analysis - Identify tasks
Chap 3.3.4	Mission, job, task analysis – Training gaps
Chap 3.4	Analysis - Task selection
Chap 3.5	Analysis - Training task analysis
Chap 3.6	Analysis - Training strategy
Chap 7	Data Model
Chap 7.3	Data model - Units of functionality
Chap 7.3.1	Unit of functionality - Aggregated Element
Chap 7.3.3	Unit of functionality - Breakdown Structure
Chap 7.3.12	Unit of functionality - Environment Definition
Chap 7.3.13	Unit of functionality - Hardware Element
Chap 7.3.14	Unit of functionality - Job Duty
Chap 7.3.19	Unit of functionality - Mission Definition
Chap 7.3.21	Unit of functionality - Part Definition
Chap 7.3.22	Unit of functionality - Performance Parameter
Chap 7.3.23	Unit of functionality - Product
Chap 7.3.25	Unit of functionality - Product Usage Phase
Chap 7.3.29	Unit of functionality - Software Element
Chap 7.3.30	Unit of functionality - Subtask Train Prioritization
Chap 7.3.32	Unit of functionality - Task
Chap 7.3.33	Unit of functionality - Task Knowledge Skill and Attitude
Chap 7.3.35	Unit of functionality - Task Resource
Chap 7.3.37	Unit of functionality - Task Train Prioritization
Chap 7.3.45	Unit of functionality - Warning Caution Train Prioritization
Chap 7.3.46	Unit of functionality - Zone Element

S3000L	International procedure specification for Logistics Support Analysis (LSA)
SX001G	Glossary for the S-Series IPS specifications

1 General

1.1 Introduction

This chapter defines the most important business objects and data elements captured during the training analysis process described in Chap 3.1 through Chap 3.6.

Business objects and data elements common to other S-Series IPS specifications are defined with a reference to SX001G, which is the S-Series IPS specifications glossary that enables harmonization and interoperability across all S-Series IPS specifications.

1.2 Scope

The following types of data elements are within scope for the S6000T training analysis activities. Data elements that:

- identify the Product to be supported
- identify items within the Product that requires training
- identify missions and user interaction with the Product
- identify performance parameters for human interaction
- identify jobs, duties, and duty tasks
- identify training gaps
- identify and describe tasks, subtasks, and subtask steps to be considered for training
- define performance requirements for a task, subtask and/or subtask step, respectively
- define Knowledge, Skill, and Attitude (KSA) requirements associated with the defined tasks to meet the defined performance requirements

1.3 Out of scope

This chapter does not give all the details that are defined in the data model (refer to Chap 7) but highlight the most important business objects and data elements that needs to be defined as part of the training analysis activities.

2 Business objects and data elements

Business objects supporting the S6000T training analysis process are organized in accordance with the respective training analysis chapter. The purpose for defining business objects is to either reflect something that exists in the real world, or just define a placeholder for information that belongs together from a logical standpoint. Data elements then define the type of data that can be captured for the respective business object.

Each business object has a reference to the Units of Functionality (UoF) in Chap 7 where complete definitions of its data elements (characteristics) are defined in the context of the S6000T data model.

2.1 Training situational analysis

The objective for the Training Situational Analysis (TSA) is to take a broad look at the need for training and to identify and evaluate possible alternative training solutions (refer to Chap 3.2).

2.1.1 Business objects - Training situation analysis

There are currently no business objects in the S6000T data model that directly document the results from the TSA.

2.2 Product information

The starting point for all Integrated Product Support analysis activities is to identify the Product to be supported (refer to Chap 3.3) along with its most important characteristics eg, its:

- Product variants
- systems, functions, hardware, etc

Note

Most of this data should have been defined during the Logistics Support Analysis (LSA), refer to S3000L.

2.2.1 Business objects - Product

Business objects and data elements which define the Product in scope are listed in Table 2.

Table 2 General business objects and data elements

Term	Definition
Product *(Business object)*	Product represents a family of items which share the same underlying design purpose(refer to SX001G). **Examples** - Airbus 340 - Pegasus engine **Reference** - Chap 7.3.23 UoF Product
Product identifier *(Data element)*	Product identifier is an identifier that establishes a unique designator for a Product and to differentiate it from other instances of Product (refer to SX001G).
Product name *(Data element)*	Product name is a name by which the Product is known and can be easily referenced (refer to SX001G).
Product variant *(Business object)*	Product variant defines a member of a Product which is configured for a specific purpose and is made available to the market (refer to SX001G). **Examples** - Boeing 787-800 vs. 787-900 - Ford Fusion S vs. SE vs. SEL **Reference** - Chap 7.3.23 UoF Product
Product variant identifier *(Data element)*	Product variant identifier is an identifier that establishes a unique designator for a Product variant and to differentiate it from other instances of Product variant (refer to SX001G).
Product variant name *(Data element)*	Product variant name is a name by which the Product variant is known and can be easily referenced (refer to SX001G).

2.2.2 Business objects - Product breakdown structure

Business objects and data elements which define the Product breakdown in scope are listed in Table 3.

Table 3 Product breakdown structure business objects and data elements

Term	Definition
Breakdown *(Business object)*	Breakdown identifies a specific partitioning of a Product to form a parent-child structure of related instances of breakdown element (refer to SX001G). **Reference** – Chap 7.3.3 UoF Breakdown Structure
Breakdown type *(Data element)*	Breakdown type is a classification that identifies the perspective from which the breakdown is defined (refer to SX001G). **Examples** – Physical breakdown – System breakdown – Functional breakdown – Zonal breakdown
Breakdown element *(Business object)*	Breakdown element defines a partition of a Product (refer to SX001G). **Note** A breakdown element can be either a hardware element, software element, zone element or an aggregated element. **Reference** – Chap 7.3.1 UoF Aggregated Element – Chap 7.3.3 UoF Breakdown Structure – Chap 7.3.13 UoF Hardware Element – Chap 7.3.29 UoF Software Element – Chap 7.3.46 UoF Zone Element
Breakdown element identifier *(Data element)*	Breakdown element identifier is an identifier that establishes a unique designator for a breakdown element and to differentiate it from other instances of breakdown element (refer to SX001G).
Breakdown element name *(Data element)*	Breakdown element name is a name by which the breakdown element is known and can be easily referenced (refer to SX001G).
Hardware element part realization *(Business object)*	Hardware element part realization is a relationship where a hardware breakdown element relates to an instance of hardware part (part as designed) which fulfills the hardware breakdown element specification (refer to SX001G). **Reference** – Chap 7.3.13 UoF Hardware Element
Part as designed *(Business object)*	Part as designed represents the definitional information for an artifact fulfilling a set of requirements, which can be produced or realized (refer to SX001G). **Reference** – Chap 7.3.21 UoF Part Definition
Part identifier *(Data element)*	Part identifier is an identifier that establishes a unique designator for a part (part as designed) and to differentiate it

Term	Definition
	from other instances of part (part as designed, refer to SX001G).
Part name *(Data element)*	Part name is a name by which the part (part as designed) is known and can be easily referenced (refer to SX001G).

2.3 Mission analysis

The objective for the mission analysis (refer to Chap 3.3.1) is to document specified, implied, and supporting missions (operational scenarios) for a Product and to identify mission tasks that will be carried out by the Product users.

2.3.1 Business objects - Mission analysis

Business objects and data elements defined during mission analysis are listed in Table 4.

Table 4 Mission analysis business objects and data elements

Term	Definition
Mission definition *(Business object)*	Mission definition represents the defining information for a Product operational scenario (refer to SX001G). **Reference** – Chap 7.3.19 UoF Mission Definition
Mission definition name *(Data element)*	Mission definition name is a name by which the mission definition is known and can be easily referenced (refer to SX001G).
Mission definition function *(Business object)*	Mission definition function represents a specific situation during the mission where a person, or group of persons, must interact with the Product. **Reference** – Chap 7.3.19 UoF Mission Definition
Product usage phase *(Business object)*	Product usage phase represents a distinct period of time during which a Product is in an operational state that has specific characteristics and requires special considerations (refer to SX001G). **Reference** – Chap 7.3.25 UoF Product Usage Phase
Product usage phase name *(Data element)*	Product usage phase name is a name by which the Product usage phase is known and can be easily referenced (refer to SX001G).
Performance parameter *(Business object)*	Performance parameter represents a metric that if changed will have a major impact on the system performance, schedule, cost and/or risk (refer to SX001G). **Reference** – Chap 7.3.22 UoF Performance Parameter
Performance parameter type *(Data element)*	Performance parameter type is a classification that identifies the type of performance parameter being defined (refer to SX001G).

Environment definition *(Business object)*	Environment definition specifies the circumstances, objects, events and/or conditions by which something can be surrounded and that influence the performance of an associated item (refer to SX001G) **Reference** – Chap 7.3.12 UoF Environment Definition

2.4 Job analysis

The objective for the job analysis (refer to Chap 3.3.2) is to identify tasks which are associated with a specific job/role, and which are critical to achievement of an organization's mission or objective.

The definition of jobs, and their associated duties and tasks, must be consistent with the task analysis resulting from an LSA, eg, performed in accordance with S3000L (refer to Para 2.5).

2.4.1 Business objects - Job analysis

Business objects and data elements defined during job analysis are listed in Table 5.

Table 5 Job analysis business objects and data elements

Term	Definition
Job definition *(Business object)*	Job definition represents a collection of homogeneous tasks, related by similarity of knowledge and skill requirements, and can be assigned to a person. **Reference** – Chap 7.3.14 UoF Job Duty
Job definition name *(Data element)*	Job definition name is a name by which the job is known and can be easily referenced.
Duty definition *(Business object)*	Duty definition defines a functional responsibility within a defined job definition. **Reference** – Chap 7.3.14 UoF Job Duty
Duty definition name *(Data element)*	Duty definition name is a name by which the duty is known and can be easily referenced.
Job task *(Business object)*	Job task is an association between the defined job or duty and a task that provides detailed information on how the job, duty, or portions thereof, will be carried out. **Reference** – Chap 7.3.14 UoF Job Duty

2.5 Identify tasks

The objective for the identify tasks analysis (refer to Chap 3.3.3) is to establish a baseline of tasks which will be analyzed from a training need perspective. User tasks identified during mission and job duty analysis activities needs to be documented as part of the training analysis process. However, tasks needed for product support should have been documented during preceding analysis activities like S3000L.

Note

Tasks originating from LSA (S3000L) are always defined in the context where requirement for the task has been identified eg, a system, a function, or a part (refer to Para 2.1).

2.5.1 Business objects - Identify tasks

Business objects and data elements defined during identify task analysis are listed in Table 6.

Table 6 Identify task business objects and data elements

Term	Definition
Task *(Business object)*	Task represents the specification of work to be done or undertaken (refer to SX001G). **Reference** – Chap 7.3.32 UoF Task
Task identifier *(Data element)*	Task identifier is an identifier that establishes a unique designator for a task and to differentiate it from other instances of task (refer to SX001G).
Task name *(Data element)*	Task name is a name by which the Task is known and can be easily referenced (refer to SX001G).
Task information code *(Data element)*	Task information code is a classification that identifies the main purpose for the Task (refer to SX001G).
Task personnel safety criticality *(Data element)*	Task personnel safety criticality is a classification that identifies the most serious health aspects that the performance of the task can pose on personnel performing the task.
Task product integrity criticality *(Data element)*	Task product integrity criticality is a classification that identifies if failure to accomplish it would result in adverse effects on the Product or system reliability, efficiency, effectiveness, safety, or cost.
Task operability impact *(Data element)*	Task operability impact is a classification that indicates the operational status and mission readiness of the end item during the task.
Task inadequate performance consequence *(Data element)*	Task inadequate performance consequence is a description that provides further information about undesired adverse reliability, efficiency, effectiveness, safety, or cost effects related to the Product or system if the accomplishment of the task fails.
Task duration *(Data element)*	Task duration is a property that specifies the average time required for the performance of a task, regardless of the number of personnel working simultaneously.
Task material resource *(Business object)*	Task material resource identifies parts which are required as a resource. **Note** Task material resource includes eg, support equipment, spare parts, and consumables. **Reference** – Chap 7.3.35 UoF Task Resource

Term	Definition
Task personnel resource *(Business object)*	Task personnel resource specifies the manpower required as a resource (refer to SX001G). **Reference** – Chap 7.3.35 UoF Task Resource
Subtask *(Business object)*	Subtask represents the specification of a work step that will be performed as part of a task (refer to SX001G). **Reference** – Chap 7.3.32 UoF Task
Subtask identifier *(Data element)*	Subtask identifier is an identifier that establishes a unique designator for a subtask and to differentiate it from other instances of subtask (refer to SX001G).
Subtask name *(Data element)*	Subtask name is a name by which the subtask is known and can be easily referenced (refer to SX001G).
Warning and caution *(Business object)*	Warning and caution define advice concerning safety, legal and health aspects (refer to SX001G). **Reference** – Chap 7.3.32 UoF Task

2.6 Identify training gaps

The objective for the identify training gaps activity (refer to Chap 3.3.4) is to identify gaps between behaviors, conditions, and standards requirements of a new or revised task and any pre-existing learning objectives included in a training curriculum.

2.6.1 Business objects - Identify training gaps

Business objects and data elements defined during identify training gaps analysis are listed in Table 7.

Table 7 Training gap analysis business objects and data elements

Term	Definition
Training gap *(Business object)*	Training gap defines a gap between the behavior, conditions, and standards required to perform a new or changed task and any pre-existing learning objectives included in a training curriculum. **Reference** – Chap 7.3.40 UoF Training Gap
Training gap identifier *(Data element)*	– Training gap identifier is an identifier that establishes a unique designator for a training gap and to differentiate it from other instances of training gap.
Training gap domain *(Data element)*	Training gap domain is a classification that determines to which learning domain the training gap is written
Training gap name *(Data element)*	Training gap name is a name by which the training gap is known and can be easily referenced.
Training gap description *(Data element)*	– Training gap description is a description that gives more information about the defined training gap.

Term	Definition
Training gap justification *(Business object)*	– Training gap justification identifies the source from which the need for additional training has been identified.

2.7 Task selection

The objective for the task selection activity (refer to Chap 3.4) is to select those tasks, subtasks and subtask steps that require training and to define its training prioritization.

2.7.1 Business objects - Task selection

Business objects and data elements that document the result from the task selection activity are listed in Table 8.

Table 8 Task selection business objects and data elements

Term	Definition
Task performance objective *(Business object)*	Task performance objective defines the precise statement of the performance expected in the execution of a task expressed in terms of the task conditions, task behavior and task standard. **Reference** – Chap 7.3.34 UoF Task Performance Objective
Task performance condition *(Data element)*	Task performance condition is a description that expresses the environmental, physical, and psychological circumstances under which the task will be performed.
Task performance behavior *(Data element)*	Task performance behavior is a description that expresses the main purpose for the task
Task performance standard *(Data element)*	Task performance standard is a description that expresses the expected quality and/or time constraints that must be met in the performance of the task.
Task train decision *(Business object)*	Task train decision defines how a task will be managed from a training perspective. **Reference** – Chap 7.3.37 UoF Task Train Prioritization
Task target audience *(Data element)*	Task target audience is a classification that identifies the kind of personnel that will be trained to perform the task.
Task human interaction requirement *(Data element)*	Task human interaction requirement is a classification that identifies the type of human coordination that needs to take place when performing the task. **Examples** – Individual task – Collective task
Task difficulty category *(Data element)*	Task difficulty category is a classification that identifies the complexity of a task from a human performance perspective.

Term	Definition
Task importance category *(Data element)*	Task importance category is a classification that identifies possible adverse effects that the performance of the task can have with respect to cost (damage) and availability for the item under analysis.
Task frequency category *(Data element)*	Task frequency category is a classification that identifies how often a Task is performed
Task training level *(Data element)*	Task training level is a classification that identifies the priority for training for the performer to be able to carry out the task within its defined performance standard. **Note** Training level is often referred to as training priority.
Task training difficulty *(Data element)*	Task training difficulty is a classification that identifies the complexity involved in training for the task.
Task training level decision *(Data element)*	Task training decision is a classification that identifies the final decision on task training level based on the task difficulty, importance, and frequency analysis as well as additional considerations such as task training difficulty, customer stipulations and regulations.
Subtask train decision *(Business object)*	Subtask train decision defines how a subtask within a task will be managed from a training perspective. **Reference** – Chap 7.3.30 UoF Subtask Train Prioritization **Note** Data elements contained in the subtask train decision are by and large identical to the ones defined for task train decision
Subtask train step *(Business object)*	Subtask train step specifies an action within a subtask that must be addressed from training perspective. **Reference** – Chap 7.3.30 UoF Subtask Train Prioritization
Subtask train step *(Data element)*	Subtask train step identifier is an identifier that establishes a unique designator for a subtask train step and to differentiate it from other instances of subtask train step.
Subtask train step name *(Data element)*	Subtask train step name is a name by which the subtask train step is known and can be easily referenced.
Warning caution train decision *(Business object)*	Warning caution train decision defines how a warning or caution will be managed from a training perspective. **Reference** – Chap 7.3.45 UoF Warning Caution Train Prioritization

2.8 Training task analysis

The objective for the training strategy activity (refer to Chap 3.6) is to identify a high-level strategy for achieving the knowledge, skills, and attitudes which have been identified for the tasks that require training.

2.8.1 Business objects - Training task analysis

Business objects and data elements that document the results from the training task analysis activity, are listed in Table 9.

Table 9 Training task analysis business objects and data elements

Term	Definition
Task knowledge level of learning *(Business object)*	Task knowledge level of learning defines the task's level of sophistication with respect to mental, cognitive, and/or logical aspects. **Reference** – Chap 7.3.33 UoF Task Knowledge Skill and Attitude
Task skill level of learning *(Business object)*	Task skill level of learning defines the task's level of sophistication with respect to physical, mechanical, and/or movement-based aspects. **Reference** – Chap 7.3.33 UoF Task Knowledge Skill and Attitude
Task attitude level of learning *(Business object)*	Task attitude level of learning defines the task's level of sophistication with respect to emotional, motivational and/or social aspects. **Reference** – Chap 7.3.33 UoF Task Knowledge Skill and Attitude

2.9 Training strategy

The objective for the training strategy activity (refer to Chap 3.6) is to identify a high-level strategy for achieving the knowledge, skills, and attitudes which have been identified for the tasks that require training.

2.9.1 Business objects - Training strategy

There are currently no business objects in the S6000T data model that directly documents the results from training strategy activity.

Chapter 4

Design

Table of contents

Chapter 4.1

Design - Introduction

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.4	Analysis - Task selection
Chap 4.2	Design - Target audience description
Chap 4.3	Design - Learning gaps
Chap 4.4	Design - Learning objectives
Chap 4.5	Design - Assessment strategy
Chap 4.6	Design - Instructional strategy
Chap 4.7	Design - Media selection
Chap 4.8	Design - Learning objective sequencing
Chap 4.9	Design - Training system alternatives

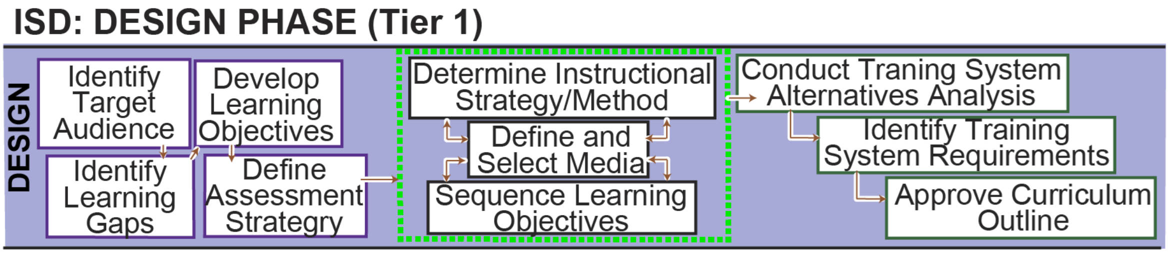
Chap No./Document No.	Title
Chap 4.10	Design - Training system requirements
Chap 4.11	Design - Curriculum outline approval and release
Chap 4.12	Business objects

1 General

During the design phase, the instructional systems designer determines how to train a specific target audience to maintain or operate a Product. The result of this phase is a training curriculum that fully describes how gaps in trainees' Knowledge, Skill, and Attitude (KSA) will be addressed in the delivery of instruction. Additionally, at the completion of this phase, all resource requirements needed for planning and acquiring resources, communicating requirements for training product development, and delivering the training are identified and approved for the training system.

2 Description

Training design is a systematic set of processes that defines a curriculum for a specific target audience based on the tasks selected for training during the analysis phase. This process is required to determine how trainees will learn and demonstrate the KSAs needed to perform a job that maintains or operates a Product. It is also required to determine the most effective and efficient methods, media, and sequencing for the given project's constraints and resource availability. The processes in the design phase are shown in Fig 1.



ICN-S6000T-B6865-00015-002-01

Fig 1 Design phase processes

The processes are listed in a sequence to aid in understanding their relationships within the design phase. In practice, the processes can be performed sequentially, as shown. However, depending on the project, some can be iterated several times, or completed in a different sequence. Various factors can affect the sequence or scope of the processes used for a project. Typical factors include but are not limited to:

- customer requirements
- budget and schedule constraints
- complexity of the training system
- availability of information

Another factor is whether the training is completely new or is a revision to existing training. For example, revising training to address a modification to an existing Product does not necessarily require a new target audience description.

The selection of processes that will be used, and their sequence or iteration, should be documented in the project's management plan. The design phase processes are described in Para 2.1 thru Para 2.10.

2.1 Target audience description

This process defines and describes characteristics of the people who will participate in the training. This information is used in later design processes to ensure effective training methods and media are selected for use in the training. Refer to Chap 4.2.

2.2 Learning gaps

This process defines and describes trainees' current KSA in relation to the KSA required to perform the tasks that were selected and prioritized for training (refer to Chap 3.4). Its purpose is to optimize what needs to be trained, and to focus the design of Learning Objective (LO) statements. Refer to Chap 4.3.

2.3 Learning objectives

This process results in LO statements being written for each learning gap. An LO is a precise statement of a learning outcome. It includes the behavior a trainee is expected to demonstrate, the conditions under which that behavior will be performed, and the degree of measurement based on a standard of performance. Refer to Chap 4.4.

2.4 Assessment strategy

This process results in the requirements for measuring and asserting the target audience's attainment of LOs. It describes the comprehensive assessment plan for testing each expected learning outcome, including the requirements and rationale for testing each LO. Refer to Chap 4.5.

2.5 Instructional strategy

This process results in how trainees will achieve the outcomes defined by the LOs. It identifies the training method(s), which is the type of activity used to impart the required KSA for each LO. Refer to Chap 4.6.

2.6 Media selection

This process determines the most efficient and effective ways to deliver learning content to trainees. The purpose of this process is to select media and define the level of fidelity required to enable the methods defined in the instructional strategy. The main consideration in media selection is the ability to enable the methods identified and effectively facilitate the learning process. Refer to Chap 4.7.

2.7 Learning objective sequencing

This process results in the chronological progression of LOs for training. It identifies the grouping and ordering of LOs for a course(s) and its component course elements (eg, training modules and lessons) to define a comprehensive training curriculum for the target audience. Refer to Chap 4.8.

2.8 Training system alternatives

This process defines optional ways to enable the training system's design based on program constraints. It considers the resources necessary for various options in relation to a curriculum outline's defined courses and course elements, the identified training methods and media, and the sequencing across the overall training delivery timeline. This process describes alternatives for achieving training system outcomes where the curriculum outline's resources are a concern or risk to successful development and delivery. Refer to Chap 4.9.

2.9 Training system requirements

This process describes all resource requirements for the development and delivery of a curriculum outline. It identifies what is needed in terms of project requirements (ie, schedule, budget, and logistical) and functional requirements (eg, technical specifications) for each resource included in the training system. This process results in the description of training system requirements to a level of detail needed for planning and acquiring resources, communicating requirements for training product development, and delivering the training. Refer to Chap 4.10.

2.10 Curriculum outline approval and release

This process places the Product training curriculum under version control and formalizes the plan of action for developing and delivering it as part of the training system. It describes the approval of the instructional design and training system resource requirements described in the curriculum outline. This process results in the release of a customer-approved curriculum outline for planning and acquiring resources, communicating requirements for training product development, and delivering the training. Refer to Chap 4.11.

2.11 Business objects

The S-Series IPS specifications use a database, described by a data model, to capture the results of the design phase in business objects. Refer to Chap 4.12.

Chapter 4.2

Design - Target audience description

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.2	Analysis - Training situation analysis
Chap 3.3	Analysis - Mission, job, and task analysis
Chap 3.5	Analysis - Task analysis
Chap 3.6	Analysis - Training strategy
Chap 4.3	Design - Learning gaps

Chap No./Document No.	Title
ASD-STE100	Simplified Technical English

1 General

To gain understanding of the training's target audience, one must collect information about that group. The target audience description process aims to define and describe the characteristics of the people who will participate in the training. This information is used in later design processes to ensure effective training methods and media will be selected for use in the training.

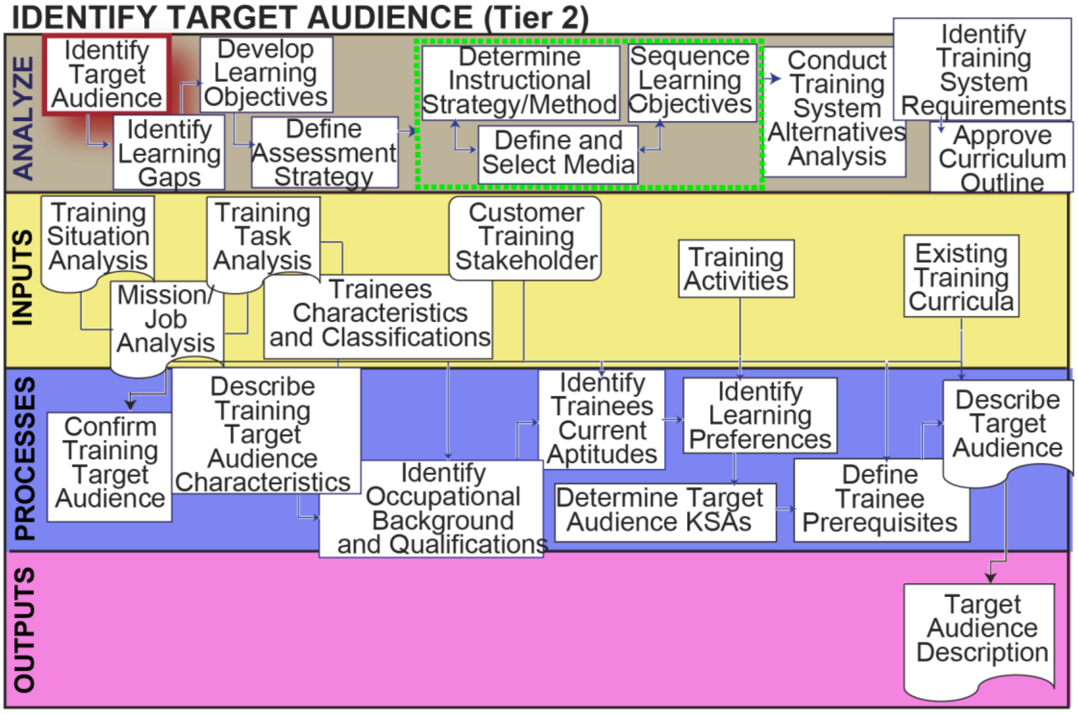
2 Description

The target audience is the group of persons to be trained. Analyzing and describing the target audience gives critical data that informs subsequent processes. Analyzing the Knowledge, Skill, and Attitude (KSA) levels of the target audience enables instructional designers to tailor the scope and complexity of learning objectives, building on trainee's current skills, and challenging them to reach new levels of achievement. Target audience data also contribute to development of effective learning strategies, methods, and media, as well as appropriate planning and scheduling decisions.

The goal of the target audience description is to reduce the vast number of individual trainee characteristics to a single set of general characteristics. This makes instructional planning feasible through the identification of common trainee's characteristics that have a direct impact on the instructional design. In other words, a target audience analysis identifies the important traits of a group, as well as identifying the range of KSA within that group that are relevant to the instructional situation.

3 Process

Fig 1 shows where, in the analysis phase, the process for describing target audience is applied.



ICN-S6000T-B6865-00016-002-01

Fig 1 Target audience description process

3.1 Inputs

The inputs to this process include, but are not limited to:

- Training Situation Analysis (TSA) process output, refer to Chap 3.2
- mission, job, and task analysis process output, refer to Chap 3.3
- task analysis process output, refer to Chap 3.5
- trainee's characteristics and classifications, refer to Para 3.1.1
- customer stakeholders, refer to Para 3.1.2
- trainee's activities, refer to Para 3.1.3
- existing training curricula, refer to Para 3.1.4

3.1.1 Trainees' characteristics and classifications

The data related to the training target audience can be gathered through different sources, which include but are not limited to:

- interviews with a sample of the target audience and/or managers
- human resources data
- customer surveys

3.1.2 Customer training stakeholders

Stakeholders within the customer's training program management, senior training leadership, and Subject Matter Expert (SME) organizations will contribute to determining the critical factors and items for consideration in the analysis of the training target audience.

3.1.3 Trainees' activities

Information about the current occupational activities and interests of the training target audience will help in identifying important characteristics for consideration in the analysis.

3.1.4 Existing training curricula

An analysis of existing training curricula learning objectives will help in developing a profile of the target audience's KSA. Depending on where the training is located within the target audience's curriculum path, entry-level or prerequisite characteristics can be determined.

3.2 Process steps

The target audience description process steps are:

- confirm the target audience, refer to Para 3.2.1
- describe training target audience characteristics, refer to Para 3.2.2
- identify occupational background and qualifications, refer to Para 3.2.3
- identify trainee's current aptitudes, refer to Para 3.2.4
- identify learning preferences, refer to Para 3.2.5
- determine target audience KSA, refer to Para 3.2.6
- define trainee prerequisites, refer to Para 3.2.7
- describe the target audience, refer to Para 3.2.8

Note

During this process, a living description is being created that can be amended or revised in later design processes.

3.2.1 Confirm the target audience

The first step is to determine who comprises the target audience, meaning the group of people to be trained. This group can include future or present job holders. Taking into consideration job information output from the mission, job, and task analysis process (refer to Chap 3.3) will be a first major step.

Also, through an understanding of the organization's Human Resources (HR) management and on-boarding process, those who will be candidates and allocated to a job that will require the training can be identified.

3.2.2 Describe the training target audience characteristics

In this step, the general characteristics of the target audience are identified, which relate to aspects such as the size, location, and availability the target population. Audience profile characteristics are also identified, which helps in describing important attributes of the target population such as their existing KSA, job history, culture, and preferences.

When researching and describing the training target audience, important considerations include, but are not limited to:

- describing characteristics of the trainees as they are, not as they should be. For example, if the trainees are generally motivated by external incentives, do not describe them as self-motivated learners.
- identifying members of the training target audience as people, not institutions or policies
- describing major differences among segments of the training target audience, as well as similarities

When identifying the general characteristics of the target audience, information that should be gathered includes, but is not limited to:

- size (ie, total number of trainees expected)
- availability of the trainees to attend the training
- throughput (ie, the average input and output of trainees expected during a given period of time)
- load (ie, the average number of enrolled trainees expected during a given period of time)
- geographical location:
    - unique site

- multiple sites
- single country
- internationally spread

When identifying audience profile characteristics, attributes of trainees within the target population that should be considered include, but are not limited to:

- age range
- gender
- type and range of educational background (include a description of the trainee's academic ability and success, if possible)
- reason(s) for attending the training
- attitude(s) about training attendance
- language(s) spoken by the trainees
- culture (eg, biases, beliefs, values)
- typical hobbies or other spare time activities
- incentives or motivators (ie, a description of trainee's motivational factors, such as a desire to seek promotion, financial factors, work-life balance, group spirit
- physical characteristics
- reading ability
- terminology or topics to be avoided
- attitude toward teamwork
- organizational memberships
- trainee's skills regarding computer tools and new technologies
- trainee's attitude towards new technologies

3.2.3 Identify occupational background and qualifications

Trainee's occupational background and employment classification should be identified. Trainees should be described by position (eg, level, rate, posting or rank). Consideration should also be given as to whether trainees have any specialization and/or will be part of a job category.

Describe any relevant competencies (ie, qualifications, certificates, licenses, employment experience) held by the trainees that can influence the training activity.

3.2.4 Identify trainee's current aptitudes

Trainee's relevant aptitudes, such as inherent capacity, talent, or ability to perform the tasks identified in the mission, job, and task analysis (refer to Chap 3.3) and training task analysis (refer to Chap 3.5) process outputs. For example, if trainees are required to possess a skill to a specific standard, this should be identified at this point. This will contribute to later determination of the training prerequisite KSA.

3.2.5 Identify learning preferences

Learning preferences can help the instructional designer determine what the target audience finds to be motivating and engaging training. Several models and inventories give explanations of how people prefer to approach or engage the learning process. For example, learning preferences are commonly described in terms of whether a person prefers learning through visual, auditory, reading, writing or kinesthetic training methods and media. Identifying these preferences can help the instructional designer create training that is more effective for or accepted by the target audience.

Researchers who have provided models for describing learning preferences include, but are not limited to:

- David Kolb
- Peter Honey and Alan Mumford
- Neil Fleming (VAK/VARK model)
- Anthony Gregorc

- Howard Gardner

3.2.6 Determine target audience KSA

In this step, the target audience's KSA are analyzed and estimated as accurately as possible, based on the information previously collected on their characteristics.

If the audience is relatively homogeneous, identify the average KSA of the target audience. If the audience is heterogeneous (ie, if an important discrepancy is noticed between groups of trainees) the instructional designer could be required to manage those groups separately and design an adaptive set of training methods and media for each group. In such cases, a different average KSA will be determined for each group.

The learning taxonomies used during the training task analysis process (refer to Chap 3.5) should be used to define an average KSA, as precisely as possible. Using these taxonomies will also ensure terminology coherency with the task analysis process output and will ease the identification of learning gaps during the perform learning analysis process (refer to Chap 4.3).

3.2.6.1 Learning taxonomies

- cognitive domain. This taxonomy will help to characterize the knowledge component of the KSA.
- psychomotor domain. This taxonomy will help to characterize the skill component of the KSA.
- affective domain. This taxonomy will help to characterize the attitude component of the KSA.

Note

All verbs used in the above taxonomies are listed in ASD-STE100 and the task analysis process (refer to Chap 3.5).

3.2.6.1.1 Example of average target audience KSA

Table 2 Example KSA for flight mechanic maintainer training

Subject	**Knowledge (cognitive)**	**Skill (psychomotor)**	**Attitude (affective)**
Hydraulic	Define the basic component of a hydraulic equipment	Replicate the tasks performed on a basic hydraulic equipment	Not applicable
Avionic	Describe the components of the avionics system	Practice removing and replacing an equipment	Demonstrate a safe behavior for replacing an avionics component
Mechanical dexterity	Recognize the mechanical dexterity requirements for the removal of a landing gear	Execute the removal of a landing gear	Recognize mechanical dexterity as an important component of Flight mechanic maintainer basic aptitudes
Technical documentation use	Identify the applicable documentation to the system	Use the applicable documentation to perform preventive maintenance	Recognize the importance of using the applicable technical documentation

3.2.7 Define trainee prerequisites

Training prerequisites are the entry level requirements the trainees must have prior to participating in the training. Prerequisites should be agreed upon and approved by the customer or stakeholders.

Prerequisites can include, but are not limited to:

- educational background
- basic knowledge
- basic skills (or can be course specific)

3.2.8 Describe the target audience

The description of the target audience includes, but is not limited to:

- occupational background and competencies
- current aptitudes
- characteristics based on the KSA
- learning preferences
- any other recommendations or special considerations

3.3 Output

The output of the target audience description process is information that gives the general characteristics and audience profile for the trainees that are within target audience, including prerequisites required of them.

Chapter 4.3

Design - Learning gaps

Table of contents

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.3	Analysis - Mission, job, task analysis
Chap 3.4	Analysis - Task selection
Chap 3.5	Analysis - Training task analysis
Chap 4.2	Design - Target audience description
Chap 4.4	Design - Learning objectives

1 General

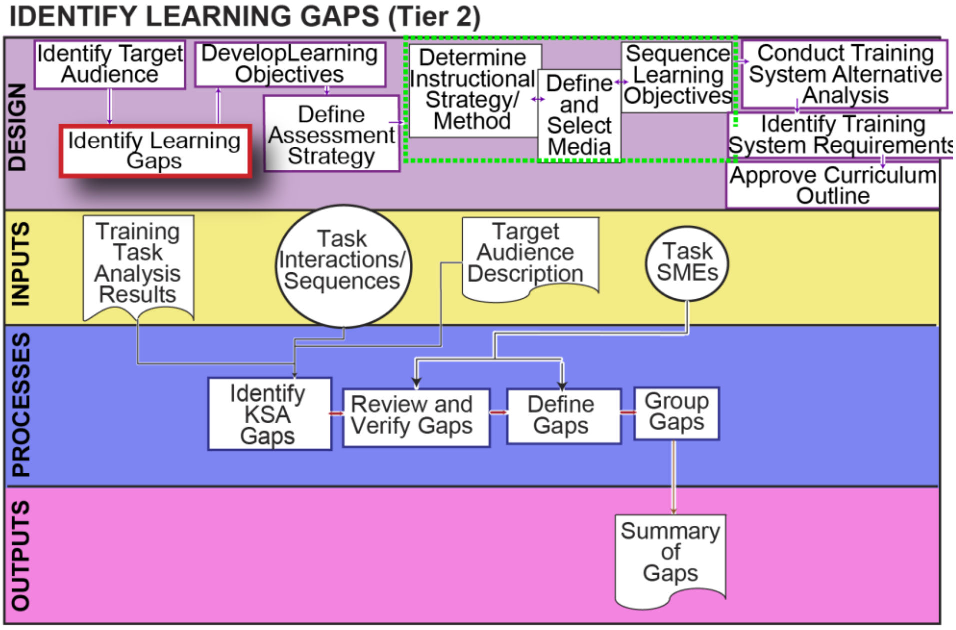
The learning gaps process is the collection, analysis, and reporting of information about trainees' current Knowledge, Skill, and Attitude (KSA) in relation to the KSA required to perform the tasks selected and prioritized for training (refer to Chap 3.4). Information concerning a trainees' characteristics and experience is compared against the requirements of the selected tasks (refer to Chap 3.5). The goal of the learning gaps process is to optimize what needs to be trained, and to focus the design of learning objective statements against these points.

2 Description

The identification of learning gaps involves the analysis of deficiencies in KSA required for the tasks to be performed versus the current KSA of the target audience. These differences between the requirements of the task and the background, credentials and experiences of the trainees are the basis for identifying the required learning objectives (refer to Chap 4.4). The learning gaps process is designed to maximize the use of resources in identifying and resolving performance deficiencies.

3 Process

Fig 1 shows the process for identifying learning gaps.



ICN-S6000T-B6865-00017-002-01

Fig 1 Learning gaps process

3.1 Inputs

The inputs to this process include, but are not limited to:

- training task analysis results, refer to Chap 3.5
- task interaction/sequences output from the mission, job, and task analysis, refer to Chap 3.3
- the target audience description, refer to Chap 4.2
- task subject matter experts, refer to Para 3.1.1

3.1.1 Task subject matter experts

Subject Matter Experts (SME) who perform the same or comparable jobs and task on similar equipment are a good source of information for identifying KSA gaps between the requirements of a task and the training a target audience will require.

3.2 Process steps

The steps in the learning gaps process are to:

- identify learning gaps, refer to Para 3.2.1
- review and verify learning gaps, refer to Para 3.2.2
- define learning gaps, refer to Para 3.2.3
- group learning gaps, refer to Para 3.2.4

3.2.1 Identify learning gaps

Learning gaps are identified between the KSA required for the tasks determined from the training task analysis (refer to Chap 3.5), and the KSA of the target audience to support the design of learning objectives (LO) (refer to Chap 4.4) required to train the target audience in the operation and maintenance of the Product being supported. The amount of information known about the target audience will improve the accuracy of learning objectives. KSA learning gaps can be addressed at multiple levels depending on the job/task requirements and the amount of information available about the target audience. Identifying gaps against each task to be performed results in the most accurate learning gap analysis.

However, it can be costly and time consuming to perform this level of analysis. An effective technique is to group tasks based on their similarity, such as by system or sub system. For example, the target audience can be evaluated against their occupational background and experience related to a braking system of a particular bicycle that is like the braking system of the new bicycle. Their competencies in the operation and maintenance of the similar braking system can be compared to the requirements of the new system and the KSA learning gaps will cover a set of tasks associated with the braking system.

3.2.2 Review and verify learning gaps

Learning gaps should be reviewed by a panel of SME and customer personnel to verify each KSA gap in relationship to the data collected about the training target audience (refer to Chap 4.2), and the training task analysis (refer to Chap 3.5). SME and customer personnel should agree on all supporting information about the target audience and tasks to be trained, and the identified learning gaps.

3.2.3 Define learning gaps

The identified learning gaps should be clearly defined. Each learning gap's definition should identify:

- whether it is a knowledge, skill, or attitude gap
- the learning taxonomy level of learning required for successful performance
- the current level of learning of the target audience

3.2.4 Group learning gaps

The identified gaps should be grouped to facilitate the development of learning objectives. This grouping should be by KSA type, as well as by task/subtask/step, as appropriate.

3.3 Output

The output of this process is a defined set of KSA learning gaps that must be addressed in the training.

Chapter 4.4

Design - Learning objectives

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.3.4	Analysis - Mission, job, task analysis - Training gaps
Chap 3.5	Analysis - Task analysis
Chap 4.2	Design - Target audience description

Chap No./Document No.	Title
Chap 4.3	Design - Learning gaps
Chap 4.5	Design - Assessment strategy
Chap 4.6	Design - Instructional strategy

1 General

A Learning Objective (LO) is a precise statement of a training outcome. It includes the behavior a trainee is expected to demonstrate, the conditions under which that behavior will be performed, and the degree of measurement based on a standard of performance.

LO statements give a foundation for the:

- assessment strategy (refer to Chap 4.5) to determine effective testing for each expected training outcome
- instructional strategy (refer to Chap 4.6) to select appropriate method(s) for training each LO

2 Description

LO statements must be written from the trainee's perspective, not the instructor's or Product perspective. They must be observable, measurable, and verifiable against their corresponding task performance objective's requirements so that the assessment of each LO can be clearly defined, and reliably measure the trainees' achievement.

An LO statement must be written to address the Knowledge, Skill, or Attitude (KSA) domain and level of learning required to address each training gap identified for a curriculum and/or learning gap identified for the target audience. The LO statement must include of the following:

- behavior
- condition(s)
- standard(s)

2.1 LO behavior

The behavior part of an LO statement defines the action a trainee will demonstrate to indicate that the KSA has been acquired. The behavior must be a verb that is observable, measurable, and appropriate to the learning domain (ie, cognitive for knowledge, psychomotor for skills, or affective for attitudes), and proficiency level required of the trainee.

Note

While it can be acceptable in some cases, it is important to avoid ambiguous verbs (eg, understand, familiarize) because it can be difficult to objectively measure such behaviors and are, therefore, difficult to reliably measure.

2.2 LO condition(s)

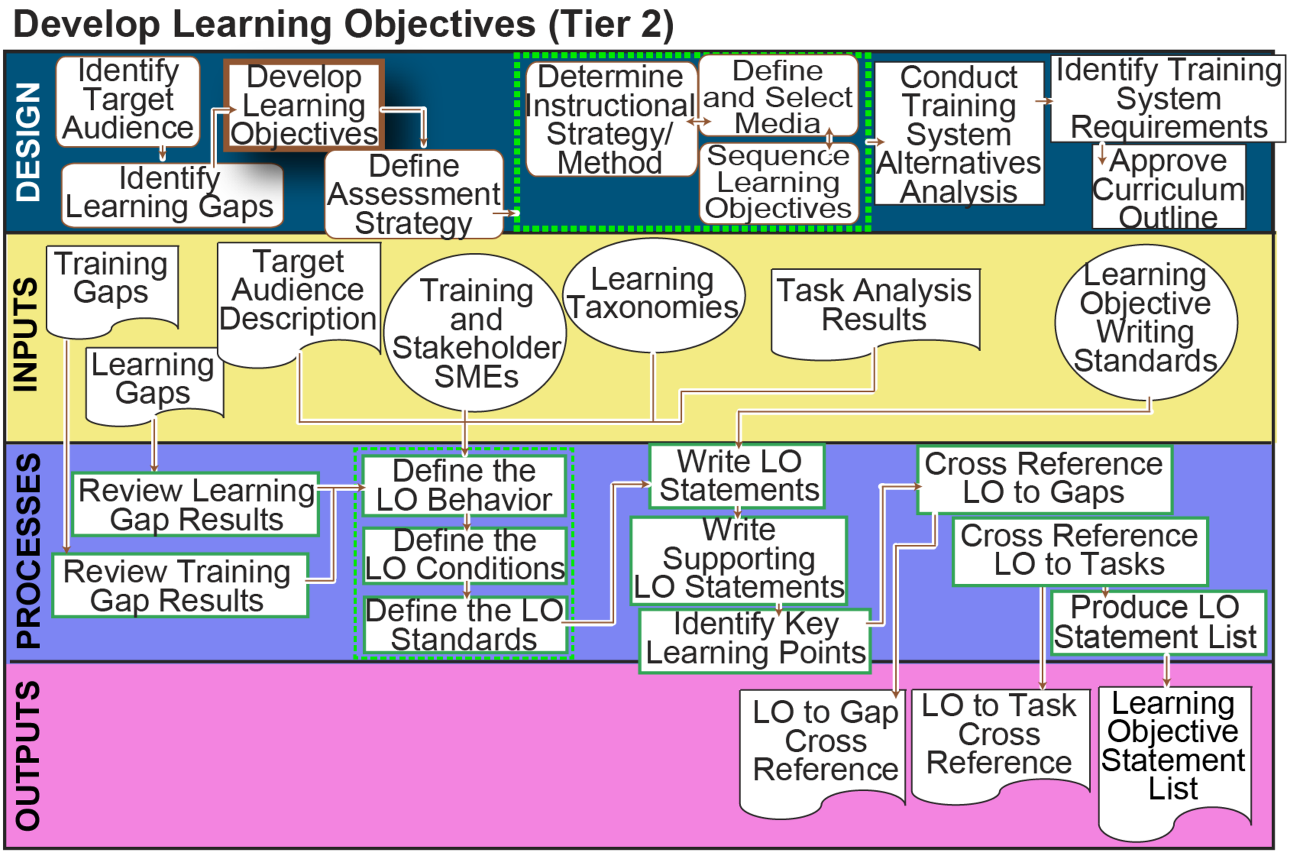
The condition part of an LO statement defines the environmental, physical, or psychological circumstances and constraints under which a behavior must be performed. Depending on project and customer requirements, the condition(s) under which a behavior is performed should reflect the actual job task conditions or the training environment. This is a project-level decision.

2.3 LO standard(s)

The standard part of an LO statement defines the criterion used to measure the trainee's performance. The standard(s) for an LO should describe the degree of proficiency that must be achieved, in accordance with any customer, authoritative, or legal regulations and processes to which the performance must adhere.

3 Process

Fig 1 shows the process for developing the learning objectives.



ICN-S6000T-B6865-00017-003-01

Fig 1 Design - Learning objectives process

3.1 Inputs

The inputs to this process include, but are not limited to:

- training gaps process output, refer to Chap 3.3.4
- learning gaps process output, refer to Chap 4.3
- target audience description process output, refer Chap 4.2
- training and stakeholder subject matter experts, refer to Para 3.1.1
- learning taxonomies, refer to Chap 3.5
- LO writing standards, refer to Para 3.1.2
- task analysis process results, refer to Chap 3.5

3.1.1 Training and stakeholder subject matter experts

At a minimum, training, and stakeholder Subject Matter Experts (SME) who are knowledgeable in the job being trained should review the learning conditions for each LO. SME can include instructors, experienced operators and maintainers, training managers, and learning standards officers. LOs should be validated by SME to ensure the behaviors, conditions and standards will give accurate, complete, and effective statements for closing trainee's KSA gaps.

3.1.2 LO writing standards

LO statements, at a minimum, will usually comprise a behavior, condition and standard. However, the project or customer can have specific LO writing guidance that includes additional requirements for creating and writing LO statements. It is important to follow any specified writing LO standards to ensure acceptance by the customer.

3.2 Process steps

The learning objectives process steps are:

- review training gaps process output, refer to Chap 3.3.4
- review learning gaps process output, refer to Chap 4.3
- define LO behavior, refer to Para 3.2.1
- define LO condition(s), refer to Para 3.2.2
- define LO standard(s), refer to Para 3.2.3
- write LO statements, refer to Para 3.2.4
- write supporting LO statements, refer to Para 3.2.5
- identify key learning points for LO statements, refer to Para 3.2.6
- cross referencing LO to gaps, refer to Para 3.2.7
- cross-reference LO to task, refer to Para 3.2.8
- produce learning objective statement list, refer to Para 3.2.9

3.2.1 Define LO behavior

The instructional designer should refer to the appropriate learning taxonomy to select a verb for an LO statement's behavior element. This should always be a verb that is measurable (eg, describe, explain, analyze), and appropriate to the learning domain and the level of learning for the KSA being trained.

Chap 3.5 gives guidance for selecting verbs within the:

- cognitive domain for knowledge
- psychomotor domain for skills
- affective domain for attitudes

Instructional designers should also consult a SME to ensure that the selected LO verbs will meet the desired training outcomes.

3.2.2 Define LO condition(s)

Once the LO behavior is identified, the instructional designer should define the condition(s) under which the training will be performed. LO conditions should be aligned with conditions described in the task analysis process results (refer to Chap 3.5). Examples of LO conditions include, but are not limited to:

- given a multimeter
- under normal operating conditions
- upon receiving a system alert
- in adverse weather conditions
- during daylight hours

If required by a project and/or customer, LO conditions can also relate to the training environment, instead of the operational or job environment. Examples of conditions that relate to the training environment include:

- in a classroom training setting
- during practical training in a learning center or school

– during supervised On-the-Job Training (OJT)

Designers should also consult a SME to ensure that the defined LO conditions are feasible and will result in the desired training outcomes.

3.2.3 Define LO standard(s)

Once the behavior and condition(s) for an LO are identified, the next step is to define the LO standard(s). LO standards give the criteria that must be adhered to when training and, subsequently, assessing trainees. Like LO conditions, LO standards can relate to the operational/job or training environment.

LO standards can be defined by, but not limited to:

- pass/fail thresholds (eg, with a minimum score of 75%)
- time allowed for the task (eg, within 30 minutes)
- training handouts (eg, in accordance with a job sheet)
- equipment operating and maintenance manuals (eg, in accordance with maintenance procedure X in equipment manual Y)
- health and safety regulations (eg, with properly worn personal protective equipment)
- electrical safety regulations (eg, in accordance with subpart X within an electrical safety standard)

Note

If no source reference is readily available for LO standards, LO standards can be based on other sources. Examples of alternate sources include, but are not limited to:

- customer business policies and procedures
- industry professional organizations (eg, National Electrical Contractors Association, Professional Aviation Maintenance Association)
- career or occupational manuals

Designers should also consult a SME to ensure that the defined LO standards are accurate and will result in the desired training outcomes.

3.2.4 Write LO statements

After the behavior, condition(s) and standard(s) of an LO are identified, the LO statement should then be written to, and associated with, the training gap and/or learning gap that the statement addresses. Each training gap and/or learning gap must have at least one LO statement associated with it.

At this point in the process steps, LO statements are often presented in a table format to make them easier to read and review. Table 2 gives an example of an LO statement in table format.

Table 2 Example LO statement in table format

Behavior	Conditions	Standards
Repair a punctured bicycle tire	In a training workshop Under supervision Using hand tools Using a repair kit	Within 80 minutes In accordance with: - Repair manual 123-4-5 - Workshop practices - Health & Safety Regulation 42

Within training course materials, a single LO statement in sentence format is also often required and should be written during this step. Using the above example, the LO statement in sentence format for bicycle technicians would be written as follows.

Repair a punctured bicycle tire under supervision in a training workshop, using hand tools and a repair kit within 80 minutes, in accordance with Repair manual 123-4-5, Workshop practices, and Health and Safety Regulation 42.

3.2.5 Develop supporting LO statements

Each LO statement is likely to contain multiple subordinate KSA that will require training. To address training requirements in a more granular way, each LO should be broken down into the appropriate number of supporting LO statements. Supporting LO statements can be developed so that each supporting KSA that is required to close a training gap and/or learning gap is fully and effectively covered.

The process steps for creating supporting LO statements are the same as described above (refer to Para 3.2.1 thru Para 3.2.4). The instructional designer should map each supporting LO to its parent LO for traceability. The method of mapping should be agreed upon with the customer and in accordance with project requirements.

3.2.5.1 Supporting LO Statement Example

Table 3 gives an example of LO statements that support the following LO statement.

Repair a punctured bicycle tire.

Table 3 Example of supporting LO statements

Behavior	Condition(s)	Standard(s)
Identify a flat bicycle tire	In training workshop Gave an example flat tire	Within 10 minutes
Remove a bicycle wheel	In a training workshop Under supervision Using hand tools	Within 15 minutes In accordance with: – Repair manual 123-4-5 – Workshop practices – Health & Safety Regulation 42

Using the example above, the supporting LO statements for bicycle technicians would be written as follows.

Identify a flat bicycle tire in a training workshop, when given an example flat tire, within 10 minutes.

Remove a bicycle wheel under supervision in a training workshop using hand tools, within 10 minutes, in accordance with Repair manual 123-4-5, Workshop practices, and Health and Safety Regulation 42.

3.2.6 Identify key learning points for LO statements

Key Learning Points (KLP) should also be identified for each LO statement. This information will assist in the design of training activities that the KSA required by the LO statement.

Note

KLP are usually only identified for LO statements that support a higher-level LO. In cases where an LO is at the highest level, and does not have supporting LO statements, KLP should be identified for that LO.

3.2.6.1 LO key learning points example

Table 4 gives an example of KLP for an LO statement.

Table 4 Example key learning points for an LO

LO statement	Key learning points
Remove a bicycle wheel	Removal of the wheel nuts Undoing the brake caliper Removal of the wheel from the bicycle frame

3.2.7 Cross - reference LO to gaps

The instructional designer should ensure there is a cross-reference between each LO statement and its corresponding training gap and/or learning gap for traceability. This gives an audit trail that can be used to determine whether a gap is addressed in the training design.

3.2.8 Cross-reference LO to task

The instructional designer should ensure there is a cross-reference between each LO statement and its corresponding task for traceability. This gives an audit trail that can be used to determine where within the training each task PO is taught and ensure that all KSA are addressed to the correct level of learning.

3.2.9 Produce LO statement list

The LO statement list contains all LO statements, including their associated KLP and cross-references to the training gap and/or learning gap they address, and the original task selected for training. If necessary, the LO statement list can be divided into multiple lists to identify different recommended training environments and/or requirements. For example, there can be individual statements for:

- formal training in a school or learning center
- on-the-job training in the workplace
- operational experience in a live environment

3.3 Output

The outputs of the learning objectives process are the LO statement list, LO to gap cross-reference, and LO to task cross-reference. The learning objective statement list gives each LO statement, including any supporting LO and KLP. The LO to gap cross-reference gives a mapping of each LO to an addressed training gap or learning gap. The LO to task cross-reference gives a mapping of each LO, including supporting LO, to its task statement. The outputs of the learning objectives process are critical inputs to the assessment strategy process (refer to Chap 4.5), and the instructional strategy process (refer to Chap 4.6).

Chapter 4.5

Design - Assessment strategy

Table of contents

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.5	Analysis - Training task analysis
Chap 3.6	Analysis - Training strategy
Chap 4.2	Design - Target audience description
Chap 4.4	Design - Learning objectives
Chap 4.6	Design - Instructional strategy
Chap 4.7	Design - Media selection

1 General

The assessment strategy process results in the requirements for measuring and asserting the trainee's achievement of Learning Objective (LO) statements. It describes the comprehensive plan for assessing each expected training outcome, including the requirements and rationale for testing each LO. How and when a trainee's achievement of LO statements will be assessed are critical inputs to the instructional strategy (refer to Chap 4.6) and media selection (refer to Chap 4.7) processes.

2 Description

An assessment strategy defines the components required for testing LO statements. There are assessment categories in which different testing types, instruments and methods can be considered when designing the assessment strategy. The selection of components will be influenced by the learning domain (ie, Knowledge, Skill, or Attitude (KSA)) and level of learning defined by each LO, and the characteristics of the target audience.

2.1 Assessment categories

Performance and knowledge are two broad categories of assessment. The assessment strategy could require one or both assessment categories to fully support the training project's LO statements.

2.1.1 Performance assessment

This assessment category measures the ability to perform a skill in a realistic task, activity, or situation. For example, a practical assessment that tests the trainee's ability to complete a system maintenance procedure is a form of performance assessment. A performance assessment can also be used to measure a trainee's attitude. For example, under simulated job conditions, an instructor can observe whether a trainee's behaviors indicate disregard for following safety protocols.

2.1.2 Knowledge assessment

This assessment category measures the comprehension of knowledge that supports the performance of a skill or attitude. For example, a theory assessment that tests comprehension of a system's general, physical, and functional capabilities is a form of knowledge assessment.

2.2 Testing types

Both assessment categories are comprised of one or more testing types. Testing types help define the placement and purpose of a given test component within the assessment strategy.

2.2.1 Pre-test/Entry test/Diagnostic test

This testing type is administered at the beginning of a major block or section of training (eg, course or lesson), prior to instruction. An example use of this test type is to assess whether trainee's current KSA meet training prerequisites and determine whether the training content should be adjusted accordingly.

2.2.2 Progress test/Formative test

This testing type assesses the KSA required by LO statements, as training progresses. How often and the points at which this testing type is used can vary based on content complexity. For example, it could be necessary to assess a trainee's comprehension of complex material after completing each section of a lesson, versus at the completion of the lesson.

2.2.3 Comprehensive test/Summative test

This testing type is administered at the end of training, or after large blocks of training. It is used to measure a trainee's mastery of the critical LO statements in a course, or to measure retention of previously learned KSA.

2.3 Testing instruments

Testing instrument selection is partly dependent on the selected assessment category, as well as the trainee's expected level of learning at course completion. For a performance assessment, the test instrument will likely be a checklist, procedure, job sheet, or similar type of instrument based on the overt behaviors a trainee is to exhibit during testing.

Knowledge assessments will a have broader range of instruments. Examples of common knowledge testing instruments include, but are not limited to:

- multiple choice. A test question characterized as an open or closed statement solved by choosing one answer from a list.
- multiple selection. A test question characterized as an open or closed statement solved by choosing one or more answers from a list.
- true/false. A test question characterized as a closed statement solved by selecting a "true" or "false" choice.
- matching. A test exercise that gives two or more lists of items to be discreetly aligned (eg, words, phrases, pictures, or symbols) in accordance with the exercise's instructions.
- completion. A test question characterized as an open-ended statement, or as a closed statement missing an element that must completed without benefit of an answer list.

2.4 Testing methods

Another aspect of the assessment strategy is the scoring method(s) used when testing a trainee's achievement of LO statements. There are two methods used to interpret the results of a trainee's performance assessment or knowledge assessment. Refer to Para 2.4.1 thru Para 2.4.2.

2.4.1 Criterion-referenced

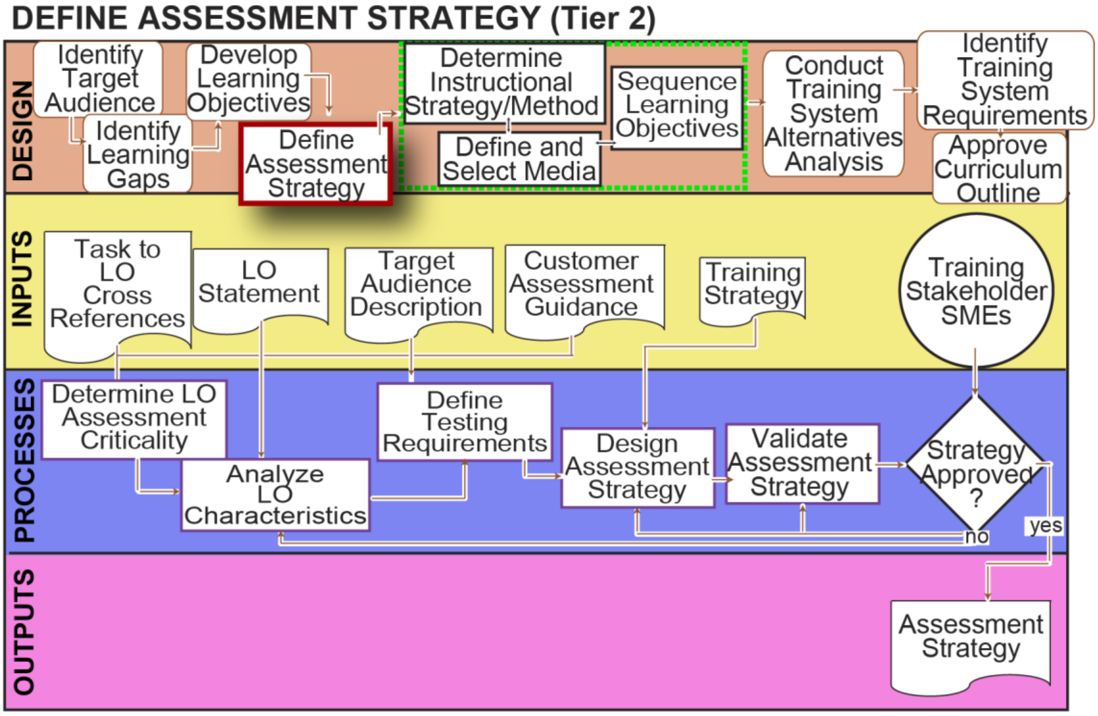
This testing method assesses a trainee based on a standard that must be attained for a successful outcome (eg, pass). Most tests use the criterion-referenced method. An example of a criterion-referenced testing method is one with a grading policy that establishes a minimum score, with no curve, for successful completion.

2.4.2 Norm-referenced

This testing method assesses a trainee based on the estimated position of the trainee's test result in a predefined population (on a curve), with respect to the KSA being measured for an LO. The importance is the relationship of the results within a group of trainees, as much as the individual trainee's result. A common example of norm-referenced testing method use is a career advancement exam.

3 Process

Fig 1 shows the process for defining the assessment strategy.



ICN-S6000T-B6865-00019-003-01

Fig 1 Design - Assessment strategy process

3.1 Inputs

The inputs to the assessment strategy process are:

- the task to LO cross-reference, refer to Chap 4.4
- the LO statement list, refer to Chap 4.4
- the target audience description, refer to Chap 4.2
- customer assessment guidance, refer to Para 3.1.1
- the training strategy, refer to Chap 3.6
- training and stakeholder Subject Matter Experts (SME), refer to Para 3.1.2

3.1.1 Customer assessment guidance

The customer can have existing policies with guidance for assessing a trainee's achievement of LO statements. Any such policies should be reviewed while designing the assessment strategy to ensure alignment with customer requirements for verifying a trainee's level of learning in the operation or maintenance of the Product.

3.1.2 Training and stakeholder subject matter experts

At a minimum, training and stakeholders SME, that are experts in the job being trained through the curriculum should review the assessment strategy. They can include, for example, instructors, training managers, and learning standards officers.

3.2 Process steps

The assessment strategy process steps are to:

- determine LO assessment criticality, refer to Para 3.2.1
- analyze LO characteristics, refer to Para 3.2.2

- define testing requirements, refer to Para 3.2.3
- design assessment strategy, refer to Para 3.2.4
- validate assessment strategy, refer to Para 3.2.5

3.2.1 Determine LO assessment criticality

In this step, the criticality of assessing each LO selected for training is analyzed to determine how it should be incorporated into the assessment strategy. It can be inefficient or of marginal value to spend resources or trainee time formally testing all LO statements. Therefore, LO assessment criticality is an important step in designing an efficient assessment strategy.

Determining the assessment criticality an LO helps define testing types that will be needed, and whether to assess LO statements through formal testing (eg, as a test score that is used in the calculation of the trainee's overall final grade) or through informal testing (eg, as an ungraded quiz that monitors the trainee's progress).

The instructional designer should use the task to LO cross-reference and LO statement list output from the learning objective process (refer to Chap 4.4) This information gives insight for determining LO assessment criticality.

3.2.1.1 Calculating LO assessment criticality

The criticality factors listed below represent the suggested minimum requirement for calculating LO assessment criticality. The customer can give additional factors that should be considered when determining criticality. LO assessment criticality factors include, but are not limited to:

- importance to performance, refer to Para 3.2.1.1.1
- frequency of use during performance, refer to Para 3.2.1.1.2

It is recommended that 1 of 3 levels be assigned to each factor as high, moderate, and low. Assign a numerical value to each level (ie, high = 3, moderate = 2, and low = 1).

Total the values assigned to each factor to determine a final score. The score will be used to group the LO statements according to their total score. The most critical LO statements should be formally tested, while least critical learning objectives can be (but not in all cases) informally tested through a variety of instructional strategies.

Table 2 gives general guidance for determining how and when an LO should be tested within the overall assessment strategy based on its calculated assessment criticality level.

Table 2 LO testing guidance based on assessment criticality level

LO Assessment Criticality Level	Testing Guidance
High	LO statements that are grouped into the upper third of criticality should be tested by a formal progress test/formative test, and formal comprehensive test/summative test.
Medium	LO statements that are grouped into the middle third of criticality should be tested by a formal progress test/formative test.
Low	LO statements that are grouped into the lower third of criticality likely only need informal testing to ensure comprehension throughout the learning process. This can be achieved, for example, through a variety of instructional strategies designed to check comprehension and give corrective feedback (as necessary), such as quizzes, short practice exercises, etc.

3.2.1.1.1 Importance to performance

Criticality to performance addresses the need for assessing an LO statement that requires a KSA that is essential to competent task performance. The more important a KSA is to successful task performance, the more critical the LO is to formally assess. The importance of an LO to performance can be identified as:

- high, value of 3. The KSA directly influences successful task performance.
- moderate, value of 2. The KSA somewhat influences successful task performance.
- low, value of 1. The KSA has little or no influence on successful task performance.

3.2.1.1.2 Frequency of use during performance

This is a measure of how often the KSA required by the LO statement is used during task performance. The more often a KSA is required, the more critical the LO is to formally assess. The frequency of using an LO during performance can be identified as:

- high, value of 3. The KSA is always/normally required during task performance.
- moderate, value of 2. The KSA is infrequently/sometimes required during task performance.
- low, value of 1. The KSA is rarely required during task performance.

3.2.2 Analyze LO characteristics

In this step, each LO statement’s behavior, condition, and standard characteristics are analyzed to ensure a reliable and valid testing mechanism is designed. The characteristics of an LO will influence the selection of the appropriate assessment category, testing instruments, and testing method. For example, if a trainee is expected to apply knowledge to discuss a system’s mission purpose, a multiple-choice test instrument would not be an effective test. A short essay would be an appropriate instrument. In contrast, if the trainee is expected to remember knowledge to identify a component in a system, then an appropriate test instrument would be a multiple-choice test.

Each LO statement’s level of learning within a domain of learning should be verified first. An LO statement’s verb gives the first indication of the required level of learning for the KSA it addresses. The condition and standard elements will then help in verifying the intended level of learning. An assessment category and testing method can then be applied to the LO.

After each LO statement’s KSA domain and level of learning is verified, the verb given as the behavior should be analyzed to determine which testing instrument(s) can accurately measure the LO statements targeted behavior. For example, if an LO requires the trainee to perform a skill, a final test consisting of multiple-choice testing instruments is inappropriate. In contrast, if an LO requires the trainee to recall the order of steps in procedure, then multiple-choice testing instruments could be appropriate.

Table 3 gives an example of general guidance for matching appropriate knowledge testing instrument types to proficiency levels in Bloom’s cognitive taxonomy (refer to Chap 3.5).

Table 3 Example matching of testing instrument to Bloom’s cognitive taxonomy

Testing Instruments	**Knowledge Remember**	**Comprehend Understand**	**Apply**	**Analyze**	**Evaluate**	**Create Synthesize**
Multiple Choice	X	X				
Multiple Selection	X	X				
True/False	X	X				
Matching	X	X				

Testing Instruments	Knowledge Remember	Comprehend Understand	Apply	Analyze	Evaluate	Create Synthesize
Completion		X	X			
Essay			X	X	X	X

3.2.3 Define testing requirements

In this step, the results from previous steps' LO assessment criticality (refer to Para 3.2.1) and LO characteristics (refer to Para 3.2.2) analyses are correlated to the prioritized list of LO statements. Each LO, or grouping of LO statements, must have an identified:

- assessment category
- testing type
- testing instrument (including alternatives, if applicable)
- testing method

During the identification of an assessment category and related testing components (ie, type, instrument, and method), the target audience description (Chap 4.2) should be considered. For example, if the target audience characteristics indicate that the trainees generally lack access to computer technology, then testing instruments that can be effectively administered in a non-digital format (ie, as an alternative to a digital format) should be included in the assessment strategy.

Additionally, any customer requirements that apply to the assessment of LO statements are identified. For example, customer organization A could require trainees to achieve a comprehensive pass rate of 75%, whereas organization B could require 90%. The assessment strategy must account for any such guidance to ensure testing methods are implemented in accordance with customer requirements. Customer assessment requirements can include, but are not limited to:

- pass or failure thresholds (eg, minimum test score)
- remediation requirements
- consequences of not meeting testing criteria or thresholds

3.2.4 Design assessment strategy

In this step, the assessment strategy is designed. The training strategy process output (Chap 3.6) should be reviewed while designing the assessment strategy to ensure requirements are in alignment, and that all components of the assessment strategy are feasible within any known schedule or budget constraints. Components of the assessment strategy should include, but are not limited to:

- assessment strategy overview. A comprehensive summary of the testing requirements from the perspective of the training context.
- test design. Details of all testing components (ie, selected categories, testing types, testing instruments, and testing methods) for each LO or grouping of LO statements. This should include justification or rationale, supported by the record of decisions made in previous steps to determine the best testing approach for each LO. It can also include how testing results will be recorded, interpreted, and acted upon.
- test development plan. Considerations for the later development of the instructional strategy to ensure alignment to and adequate trainee support for the testing components. For example, if an instructional designer knows that a summative test will emphasize the performance of highly critical handling skills, then an appropriate method (eg, simulation) for the trainee to practice those skills during training periods can be included.

Customer requirements for a complete assessment strategy should also be reviewed. Depending on the training project and customer guidance, additional information can be required in the assessment strategy.

3.2.5 Validate assessment strategy

In this step, the assessment strategy is reviewed by training stakeholders that are experts in the job being trained through the project's training curriculum. They can include, for example, instructors, training managers, and learning standards officers.

This review can result in revisions to individual LO testing requirements and/or the overall assessment strategy. Steps in this process can be revisited, if necessary.

Once training stakeholder SME agreement and approval of the assessment strategy is achieved, the process is complete.

3.3 Outputs

The output of this process is an approved assessment strategy, which gives a comprehensive plan for testing trainee's achievement of LO statements. It defines how and when each, or group of, LO statements, will be tested, including the rationale for the selected components, and testing requirements.

Chapter 4.6

Design - Instructional strategy

Table of contents

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.6	Analysis - Training strategy
Chap 4.2	Design - Target audience description
Chap 4.4	Design - Learning objectives
Chap 4.5	Design - Assessment strategy
Chap 4.7	Design - Media selection

1 General

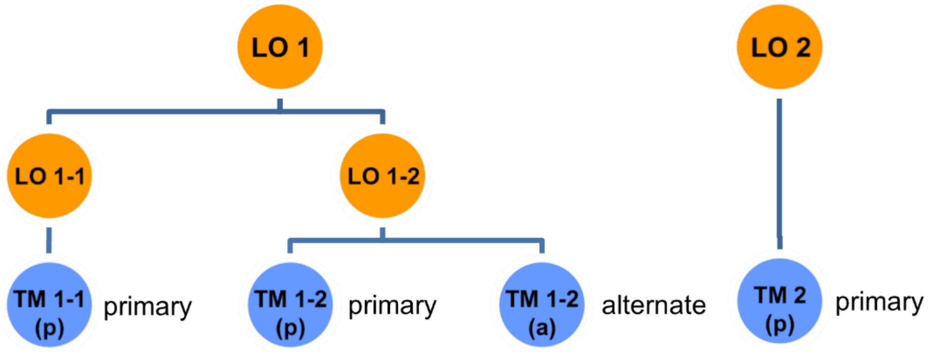
The instructional strategy process results in how trainees will achieve the outcomes defined by the Learning Objectives (LO). It identifies the training method(s), which is the type of activity used to impart the Knowledge, Skill, or Attitude (KSA) addressed by each LO. The output of the instructional strategy process gives critical input to the media selection process (refer to Chap 4.7), which identifies media that are appropriate for the selected training method(s) and target audience.

2 Description

The instructional strategy process defines a comprehensive design of training methods to facilitate the trainee's learning process. A primary training method is assigned to an LO. An alternate training method can also be assigned to an LO.

Fig 1 shows generic examples of how training methods can be assigned when an:

- LO that has one or more supporting LO (ie, LO1)
- alternate training method is selected for an LO (ie, LO 1-2)
- LO that has no supporting LO (ie, LO 2)



ICN-S6000T-B6865-00073-002-01

Fig 1 Examples of assigning a training method to an LO

Training methods (ie, how to train) are the focus of the instructional strategy process, which can include, but are not limited to:

- group-paced training
- self-paced training
- case study
- demonstration
- lecture
- role-play
- structured On-the-Job Training (OJT)
- facilitated training
- simulation

Note

Media are used to convey content during training. Training methods can often use a variety of media types to achieve the same learning outcomes. The media used for a training method selected for an LO are not identified during the instructional strategy process. The media selection process is conducted separately (refer to Chap 4.7).

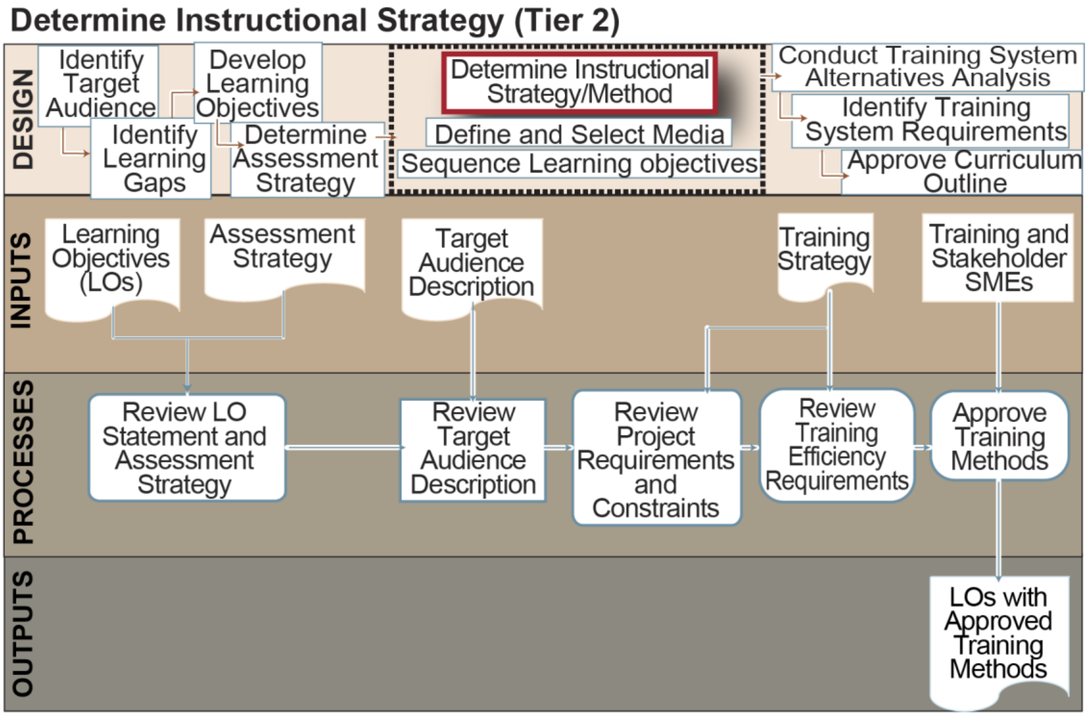
Throughout the instructional strategy process, considerations made during training method selection are based on, but not limited to the:

- LO learning domain and level of learning, refer to Chap 4.4
- LO testing requirements, refer to Chap 4.5

- target audience's intrinsic characteristics (eg, preferences, prior knowledge, experience), refer to Chap 4.2
- size, location, and availability of the target audience, refer to Chap 4.2
- context or environment in which training will be conducted, refer to Chap 3.6
- schedule and budget constraints for the training project, refer to Chap 3.6

3 Process

Fig 2 shows the instructional strategy process.



ICN-S6000T-B6865-00020-002-01

Fig 2 Instructional strategy process

3.1 Inputs

The inputs for the instructional strategy process are:

- LO, refer to Chap 4.4
- target audience description, refer to Chap 4.2
- assessment strategy, refer to Chap 4.5
- training strategy, refer to Chap 3.6
- training and stakeholder Subject Matter Experts (SME), refer to Para 3.1.1

3.1.1 Training and stakeholder subject matter experts

At a minimum, training, and stakeholder Subject Matter Experts (SME) who are knowledgeable in the job being trained should review the output of the instructional strategy process. A SME can be, but is not limited to, an instructor, experienced operator or maintainer, training manager, or learning standards officer. A SME can give practical expertise that will help in validating that the selected training methods will facilitate the learning process, are feasible, and result in trainee success.

3.2 Process steps

The instructional strategy process steps are:

- review LO statement list and assessment strategy, refer to Para 3.2.1
- review target audience description, refer to Para 3.2.2
- review training project requirements and constraints, refer to Para 3.2.3
- review training efficiency requirements, refer to Para 3.2.4
- approve training methods, refer to Para 3.2.5

3.2.1 Review LO statement list and assessment strategy

The LO statement list output from the learning objectives process (refer to Chap 4.4) and the assessment strategy (refer to Chap 4.5) are reviewed in this step. The learning domain and level of learning characteristics of each LO are reviewed to identify appropriate candidate training methods. LO assessment categories and testing instruments are also considered to ensure that training methods align to and reflect how trainees will be assessed.

Training methods have varying characteristics that enable the learning process in different ways. It is important to ensure that a candidate training method can support the condition(s) under which trainees are expected to demonstrate the behavior defined in an LO. How the trainee will practice and test for that LO should also be reviewed. For example, if an LO requires a skill performance assessment, then the candidate method(s) should have characteristics that support skill behavior performance. These considerations will help in identifying all possible methods with appropriate characteristics before making final selections.

As a final consideration in this step, any LO with a KSA that was identified as a prerequisite for the training (refer to Chap 4.4) should be verified with the customer. This can help ensure that any entry KSA required of the trainee's will be addressed and satisfied, as expected.

3.2.2 Review target audience description

The previous step resulted in the initial identification of training methods that are appropriate for each LO. The target audience description (refer to Chap 4.2) is reviewed next to refine the pool of methods for optimal training effectiveness.

Trainees can learn through different training methods. Some methods can be more effective for a given group of trainees. The instructional systems designer should consider the target audience's characteristics and preferences when refining the pool of training methods for an LO. For example, if intrinsic characteristics such as learning preferences are given in the target audience description, then that information can be used to give priority to methods that are most compatible with the target audience's preferred means of training.

A trainee's prior knowledge and experience can also influence the effectiveness of training methods. For example, training methods that can give a higher level of guidance throughout the learning process will generally be more effective for a trainee who is a novice, versus being more advanced and experienced. The instructional systems designer should take into consideration the trainee's existing KSA from relevant job and training experiences when prioritizing or selecting training methods for an LO.

The instructional systems designer should also consider the extrinsic characteristics of the trainees. For example, if trainees will be in different locations across the globe when the training is given, then training methods with characteristics that are conducive to asynchronous training (eg, self-paced training) should be considered over those that are characteristically synchronous (eg, group-paced training).

3.2.3 Review training project requirements and constraints

Additional factors that can influence the feasibility of any given method should be considered as training method selections are refined. The instructional systems designer should review the training strategy process' output (refer to Chap 3.6) for information that can eliminate any of the candidate training methods. For example, the training strategy could have a schedule constraint that identifies a date on which training will be conducted for the target audience. If the development timeline needed to develop and implement a particular method (eg, simulation) exceeds that schedule constraint, then an alternate training method (eg, demonstration) should be selected for that first training event.

Any additional budgetary, environmental, and resourcing constraints should also be reviewed. For example, if the budget required for development of a particular training method will be unavailable until later years, then an alternate method that can be afforded should be selected.

3.2.4 Review training efficiency requirements

The training strategy process' output (refer to Chap 3.6) can include additional requirements that influence a training system's initial and lifecycle costs. Such requirements often address efficiency factors to decrease training costs, while increasing the training throughput. If available, the instructional systems designer should review this information to ensure the highest probability of success for the selected training methods.

3.2.5 Approve training methods

At a minimum, the selected training methods should be reviewed and approved by training and stakeholder SME. A SME can be, but is not limited to, an instructor, experienced operator or maintainer, training manager, or learning standards officer.

This final review can result in revisions to the selection and assignment of a training method(s) to an LO. It can also result in the need to reconsider alternate training methods. Steps in this process should be revisited accordingly, as necessary.

Once SME approval of the training method(s) selected for every LO is received, the instructional strategy process is complete.

3.3 Outputs

The output of the instructional strategy process is an approved set of training methods. At least one primary training method must be assigned to each LO. The assignments should also include the rationale and requirements for each selected training method.

Chapter 4.7

Design - Media selection

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.6	Analysis - Training strategy

Chap No./Document No.	Title
Chap 4.2	Design - Target audience description
Chap 4.4	Design - Learning objectives
Chap 4.5	Design - Assessment strategy
Chap 4.6	Design - Instructional strategy
S3000L	International procedure specification for Logistic Support Analysis

1 General

The media selection process determines the most efficient and effective ways to deliver learning content to trainees. The purpose of this process is to select media and define the level of fidelity required to enable the training methods selected for learning objectives (LO). Media is the plural of medium. Regarding training environments, media are the means of communicating and transferring a learning concept to a trainee. Media are the replicable forms, or aids by which instructional content is formatted, stored, and delivered to the trainee. The main consideration in media selection is their ability to enable the training methods in ways that facilitate the learning process.

Although training methods and media are discussed in individual processes, they should be considered interdependently and can be performed simultaneously. No single medium is the most appropriate choice for every training method or situation. Proper media selection helps ensure training content is developed and delivered as efficiently and effectively as possible.

2 Description

This chapter contains guidance for optimizing the delivery format of content to trainees. The media selection process identifies effective media for training interventions, as well as the most efficient means (ie, in terms of time and cost) for delivering content. The process requires review of the training project's LO statement list (refer to Chap 4.4), instructional strategy (refer to Chap 4.6), time and budget constraints, and resource availability. It also requires knowledge of potential media choices, including their characteristics and advantages/disadvantages in supporting the trainees' learning process.

Media are the delivery mechanisms for training content, including any sensory stimuli, to enable the learning process. Media can be simple (eg, 2D line drawing) or complex (eg, mixed reality). Examples of media include, but are not limited to:

- video
- printed material
- interactive multimedia
- technical training equipment
- an immersive virtual environment
- a full-motion simulator

Note

Similar terms for interactive courseware include, interactive multimedia instruction, and computer-based training.

Media selection is often an iterative process that spans the analysis and design phases. Preliminary media selections could be identified earlier during the training strategy (refer to Chap 3.6) analysis process for media that require long design and development cycles, such as simulators and immersive virtual content. Preliminary media selection is important when acquisition and development costs are factors that influence the training strategy.

Detailed media selection is performed during the design phase to ensure the optimal media are developed to support the training methods identified for each LO. Media selection and fidelity analysis are closely tied and should be performed together. For example, if interactive courseware is identified as the instructional medium to enable a self-directed training method, then fidelity analysis can help determine the content requirements for the interactive courseware. If high fidelity is deemed unnecessary, then content consisting of 2D images could be the requirement, versus a fully interactive, and more expensive, 3D model.

Media selection can be a complicated and intricate process. A variety of media selection models are available to assist instructional designers.

2.1 Media selection models

Media characteristics and selection is an area of many research and development efforts in the field of Instructional Systems Design (ISD). This has resulted in the development of many practical models to perform the media selection process.

The application of any media selection model requires a good understanding of the training project's intended LO outcomes, training methods, trainees, available resources, and project constraints. The priority in selecting appropriate media should be their effectiveness in supporting the learning process and assessment. One medium is often not enough to give all the conditions required for effective Product training. Complex learning approaches often require multiple types of media, which requires the instructional designer to decompose media selections into more granular mixes of media components. Key factors to consider when choosing appropriate media include, but are not limited to:

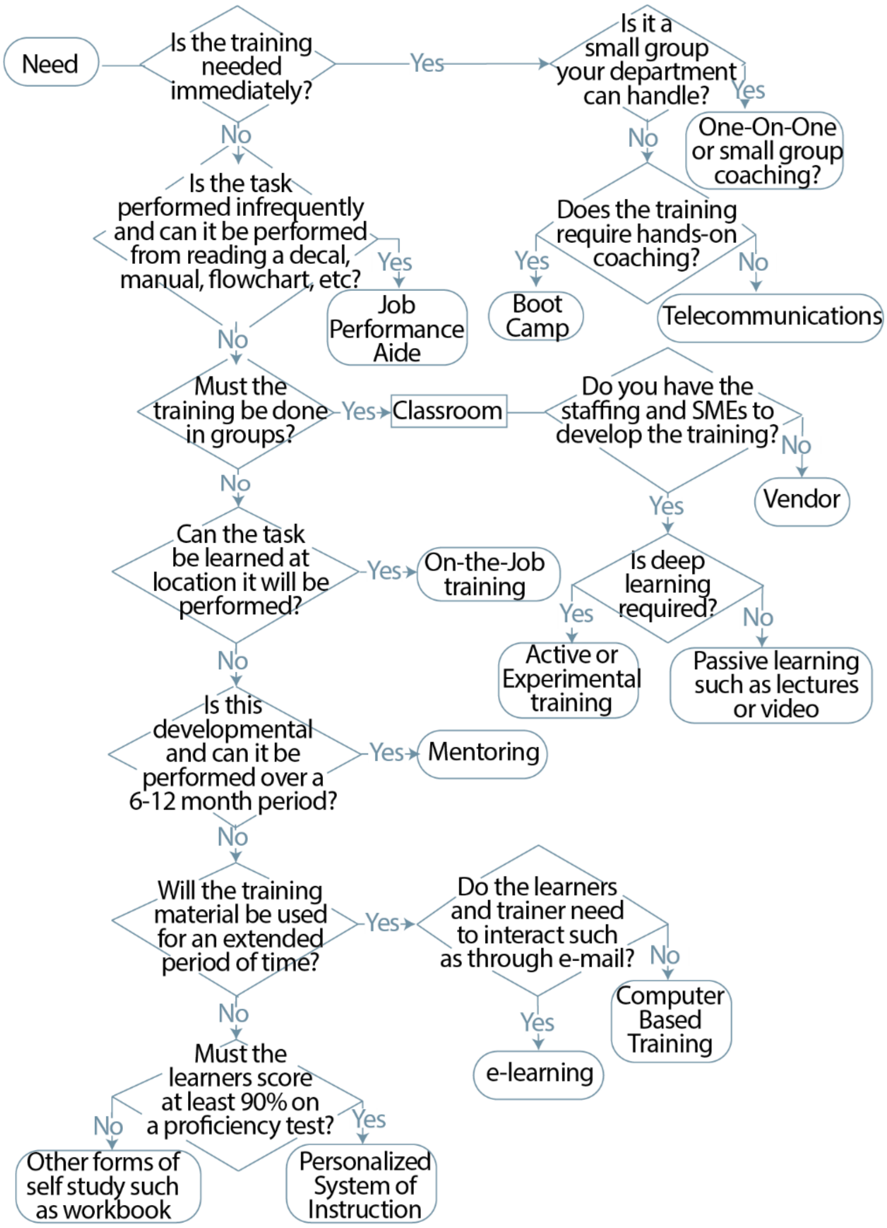
- media characteristics
  - can the medium support the desired learning outcomes, including the selected training method and necessary level of fidelity?
- trainee throughput
  - can the medium support the number and volume of trainees to be trained, including any geographical and availability considerations?
- training management
  - does the medium provide for efficient management of training?
- resource constraints
  - is the medium selected cost efficient, does it support the constraints of the training program?

Media selection models generally approach the decision process through:

- media selection flowcharts, refer to Para 2.1.1
- media selection matrices, refer to Para 2.1.2

2.1.1 Media selection flowcharts

Media selection flowcharts guide instructional designers through a series of branching decisions to identify the type(s) of media that should be selected. Process diagrams and checklists are designed to match media to the project's characteristics and instructional requirements. Decision flowcharts can address multiple levels of granularity in the media selection process. The decision points within this kind of approach usually result in a single media option being identified. Fig 1 is an example of a media selection flowchart model (Clark, R. E. (1983)).



ICN-S6000T-B6865-00074-003-01

Fig 1 Media selection flowchart example

2.1.2 Media selection matrices

Media selection matrix models give instructional designers a set of decision aids that match LO requirements to the capabilities of a potential media pool. For example, if an LO's level of learning requires application, and the target behavior to be assessed is a verbal response, then the medium should have verbal detection and identification capabilities. Matrices will often give multiple viable media, which can then be down selected based on other project requirements, if necessary. Fig 2 is an example of a media selection matrix. Media matrices can also be weighted to rate critical project factors with higher priority.

Media	Cognitive Levels of Learning						Content Presentation							Trainee Responses/ Interactions					
	Evaluation	Synthesis	Analysis	Application	Comprehension	Knowledge	Support Drawings/Diagrams	Visual Still	Support Animation	Graphics	Support Full Motion Video	Color	Audio/Verbal	Student/Instructor Dialogue	Inter-Student Dialogue	Written/Text Entry	Procedural Response	Verbal Response	Decision Indicator
IMI Level 4	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X
IMI Level 3	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X		X
IMI Level 2				X	X	X	X	X	X	X	X	X	X			X	X		X
IMI level 1						X	X	X	X	X	X	X	X						
Classroom	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Video Conferencing	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X	
Augmented Virtual Reality	X	X	X	X	X	X	X	X	X	X	X	X	X				X		X

ICN-S6000T-B6865-00022-002-01

Fig 2 Media selection matrix example

Note

Interactive Multimedia Instruction (IMI) is a generic term that can also be referred to as Interactive Courseware (ICW), or Computer Based Training (CBT).

2.2 Fidelity analysis

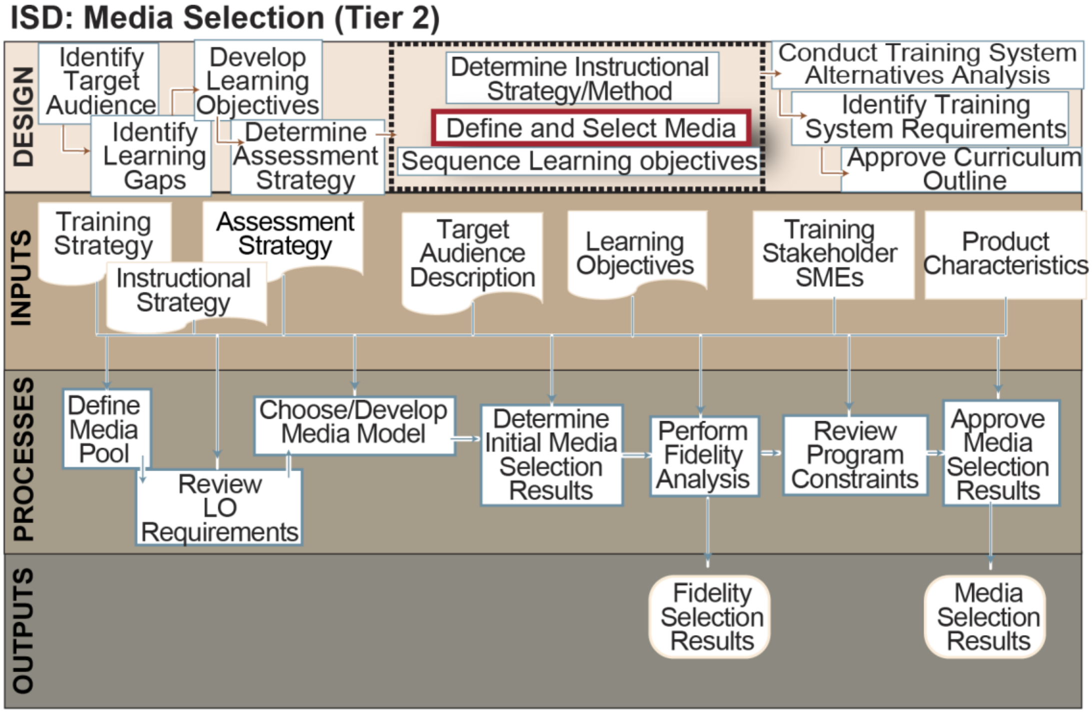
Fidelity refers to how closely a training event must imitate or replicate the real-world attributes of the Product to make the training effective. Fidelity is typically considered in terms of simulations but is applicable to all training media. Fidelity analysis enhances traditional media selection, which primarily uses the intended LO outcomes and training methods as the decision basis, and the target audience's characteristics as critical considerations in the selection process.

Fidelity analysis is focused on determining the degree to which media must mimic real-world properties to optimize the learning process and to ensure LO outcomes are achieved. When considering media fidelity, the instructional designer's considerations include, but are not limited to the:

- LO learning domains, levels of learning, conditions, and standards
- LO testing requirements
- Product's characteristics
- characteristics of the selected media
- cost of the selected media
- current knowledge and expertise of the target audience
- training program or project schedule and budget constraints

3 Process

Fig 3 shows the media selection process.



ICN-S6000T-B6865-00021-002-01

Fig 3 Media selection process

3.1 Inputs

The inputs for the media selection process include the:

- training strategy, refer to Chap 3.6
- instructional strategy, refer to Chap 4.6
- assessment strategy, refer to Chap 4.5
- target audience description, refer to Chap 4.2
- LO, refer to Chap 4.4
- training and stakeholder Subject Matter Experts (SME), refer to Para 3.1.1
- Product characteristics, refer to Para 3.1.2

3.1.1 Training and stakeholder subject matter experts

At a minimum, training and stakeholder Subject Matter Experts (SME) should review the media selections. A SME can be, but is not limited to, an instructor, experienced operator or maintainer, training manager, or learning standards officer. SME review will help ensure that the selected media will enable the training methods, are feasible, and will result in trainee success throughout the learning process and during assessment.

3.1.2 Product characteristics

Product characteristics are properties that describe the depiction of the Product. Characteristics of a Product typically include physical properties, classifications, and descriptions. Product characteristics to consider include, but are not limited to:

- size
- shape
- weight
- performance
- movement

3.2 Process steps

The media selection process steps are:

- define media pool, refer to Para 3.2.1
- review LO requirements, refer to Para 3.2.2
- choose or develop media selection model, refer to Para 3.2.3
- determine initial media selection, refer to Para 3.2.4
- perform fidelity analysis, refer to Para 3.2.5
- review program constraints, refer to Para 3.2.6
- approve media selection results, refer to Para 3.2.7

3.2.1 Define media pool

This process begins with defining the media pool, which comprises all media options the training project can consider during media selection. Media have characteristics that instructional designers use to determine whether they support the methods identified during the instructional strategy process (refer to Chap 4.6), and the testing requirements defined during the assessment strategy process (refer to Chap 4.5) for each LO. Media characteristics should be considered carefully to ensure only appropriate media options are included in the media pool. The advantages and disadvantages of the media types should also be considered.

When defining the media pool, the instructional designer should consider whether the media:

- conflicts with the specific training environment
- effectively supports the LO domain and level of learning
- enables the selected training method
- enables appropriate testing of the LO
- supports the training strategy (refer to Chap 3.6.)

An example media pool for a self-paced training method for acquiring knowledge about a Product could include:

- video
- Interactive Multimedia Instruction (IMI)
- part task trainer
- 3D model
- virtual world

3.2.2 Review LO media requirements

The behavior, condition(s) and standard(s) attributes of an LO dictate the inherent requirements for effective learning. These requirements will vary from one LO to another, and influence media selection. The following tables identify the types of requirements that the instructional designer should consider before selecting media for an LO in the:

- cognitive domain (knowledge), refer to Table 2
- psychomotor domain (skills), refer to Table 3
- affective domain (attitudes), refer to Table 4

Table 2 Example requirements for an LO in the cognitive domain (knowledge)

Requirement	Definition
Level of learning	Level within Bloom's cognitive taxonomy to which the knowledge LO is written.
Content presentation	The requirements of the content to support the LO. These include, but are not limited to: – alphanumeric – visuals

Requirement	Definition
	– still or motion graphics – solid object representation – color audiovisuals
Trainee responses/interactions	The trainee response requirements include, but are not limited to: – student-instructor dialogue – inter-student dialogue – written text entry procedural response – verbal response – decision indicator
Presentation requirements	Requirements of the presentation include, but are not limited to: – amount and type of student interactions – self-pacing or instructor pacing – free play or experimentation
Trainee assessment/feedback	Requirements for trainee assessment and/or feedback include, but are not limited to: – type of response required – immediate or delayed feedback
Instructor control	Requirements for instructor monitoring and management include, but are not limited to: – view of trainees' displays – stop/pause/repeat control – communications

Table 3 Example requirements for an LO in the psychomotor domain (skills)

Requirement	Definition
Level of learning	Level within Simpson's psychomotor taxonomy to which the skill LO is written.
Physical motion	The physical motion requirements include, but are not limited to: – acceleration – vibration buffering – pitch and roll
Tactile response	The tactile requirements include, but are not limited to: – touch/feel – physical reaction
Environmental	Environmental requirements include, but are not limited to: – lighting – temperature – ambient noise – color – 3D solid object representation

Requirement	Definition
	– resolution depth
Communication	Communication requirements include, but are not limited to: – digital/auditory communications – person to person – third-party communications
Trainee assessment/feedback	Requirements for trainee assessment and/or feedback include, but are not limited to: – type of response required – immediate or delayed feedback
Instructor control	Requirements for instructor monitoring and management include, but are not limited to: – view of trainees' displays – stop/pause/repeat control – communications

Table 4 Example requirements for an LO in the affective domain (attitudes)

Requirement	Definition
Level of learning	Level within Bloom's affective taxonomy to which the attitude LO is written.
Content presentation	The requirements of the content to support the LO. These include, but are not limited to: – alphanumeric – visuals – still or motion graphics – solid object representation – color audiovisuals
Trainee assessment/feedback	Requirements for trainee assessment and/or feedback include, but are not limited to: – type of response required – immediate or delayed feedback
Instructor control	Requirements for instructor monitoring and management include, but are not limited to: – view of trainees' displays – stop/pause/repeat control – communications

3.2.3 Choose or develop media selection model

Media selection models give a level of objectivity and rationale when choosing an appropriate medium for a particular LO and training method. The model choice depends on several different factors (refer to Para 2.1). Given the complexity of media available, as well as training requirements, a media selection model should be used to aid in deciding the most appropriate medium to use for an LO and training method. If an adequate media selection model is unavailable, the instructional designer can develop one. The chosen or developed media selection model should be tested against sample LO to verify that it supports the training system requirements. It should also be approved by the customer and/or training stakeholders.

3.2.4 Determine initial media selection results

The chosen media selection model should be applied to each LO, including its selected training method. The media selection model must recommend a primary medium for each LO. It should also recommend at least one alternate medium.

3.2.5 Perform fidelity analysis

Fidelity analysis determines how realistic a medium must be to support the requirements of an LO (refer to Para 2.2). For example, if an LO includes the performance of a procedure that requires a knob to be turned and the instructional method is simulation, fidelity analysis determines how important it is in the learning process for the trainee to receive tactile feedback when turning the knob.

The natural approach is to replicate the Product behavior as realistically as possible. Generally, however, higher media fidelity results in higher costs (ie, both initial acquisition and in sustainment). Furthermore, the highest fidelity could be unnecessary, and can result in diminished returns. For example, there could be less value returned if an interactive simulation is developed to train experienced technicians on a common battery pack replacement procedure, instead of a less costly video clip that can be just as effective. Therefore, a fidelity analysis should be performed to refine the initial results of the media selection process.

Table 5 lists and describes the different types of media fidelity.

Table 5 Types of media fidelity

Fidelity Type	Description
Physical	The realism of the media in relation to the Product, as perceived by the human senses (eg, visual, tactile). Technical training equipment, including Part Task Trainers (PTT), Full Task Trainers (FTT), and the actual equipment have a high level of physical fidelity.
Task	The correspondence between tasks performed on a Product in the real-world and the media used in the training environment. Flight simulators tend to have a high level of task fidelity.
Behavioral	The extent to which the training media must functionally act or operate like the real-world equipment. Free play simulations tend to have a high behavioral fidelity (eg, electricity and data flow).
Environmental	The degree to which the training media duplicates the external sensory conditions experienced when performing the task on equipment in a real-world setting. Training in the job setting can produce a high level of environmental fidelity.
Psychological	The degree to which the training media duplicates the cognitive or emotional conditions experienced when performing the task on equipment in a real-world setting. A simulation that requires the timely completion of a maintenance task under emergency conditions to avoid crew casualties can have a high level of psychological fidelity.
Haptic	The degree to which the training media must duplicate the sense of touch that would be felt by a user interacting with the actual Product.

3.2.6 Review program constraints

During this step, the viable media options selected for each LO is reviewed against program or project constraints defined during the training strategy process (refer to Chap 3.6). Reviewing these constraints will answer questions such as:

- is the selected media effective and the most cost-efficient?
- can the selected media be scaled to the required trainee audience size and throughput?
- is there adequate time to acquire and develop the selected media?
- will the resources necessary to support the selected media be available?
- are the selected media within the anticipated training project's budget?

Note

Cost is a critical constraint when determining the most efficient medium. Life Cycle Costs (LCC), and total ownerships costs for sustainment of the training program should be carefully considered when choosing media. For a more detailed description, as well as the cost analysis process, refer to S3000L.

These types of questions can require the instructional designer to revise the media selections to accommodate training program constraints.

3.2.7 Approve media selection results

Media selections should be reviewed by training and stakeholder SME in this step. A SME can be, but is not limited to, an instructor, experienced operator or maintainer, training manager, or learning standards officer.

This review can result in revisions to the initial media selections. If necessary, the steps in this process can be revisited, as necessary.

Once the instructional designer has training and stakeholder SME approval, the media select results are finalized, and the process is complete.

3.3 Outputs

The output of the media selection process is the media selection results, which includes a primary medium selection for each LO. It should also include at least one alternate medium for each LO method, and the rationale for selecting each primary and alternate medium.

Chapter 4.8

Design - Learning objective sequencing

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.5	Analysis - Task analysis
Chap 3.6	Analysis - Training strategy
Chap 4.2	Design - Target audience description
Chap 4.4	Design - Learning objectives
Chap 4.5	Design - Assessment strategy
Chap 4.6	Design - Instructional strategy

Chap No./Document No.	Title
Chap 4.7	Design - Media selection

1 General

The learning objective sequencing process results in the chronological progression of Learning Objectives (LO) for training. It identifies the LO groupings and ordering for a course(s) and its component course elements (eg, training modules and lessons) to define a comprehensive training curriculum for the target audience.

2 Description

The process for organizing LO statements into an optimal sequence for training requires instructional designers to make iterative refinements. These should be based on the characteristics of each LO to be trained, the target audience, and other project-specific considerations until an efficient and effective sequence is defined.

2.1 Curriculum outline

An important purpose of curriculum is to map the sequence of LO statements for training. The focus of the learning objective sequencing process is to define how LO statements are organized into courses and course elements in relation to each other. The result is a draft curriculum outline, including training time estimates.

The major activities of this process are:

- LO grouping, refer to Para 2.1.1
- LO ordering, refer to Para 2.1.2

Once complete, the LO sequence for a curriculum is fully defined. The resulting curriculum outline gives a comprehensive blueprint of where each LO will be addressed within the training for the target audience.

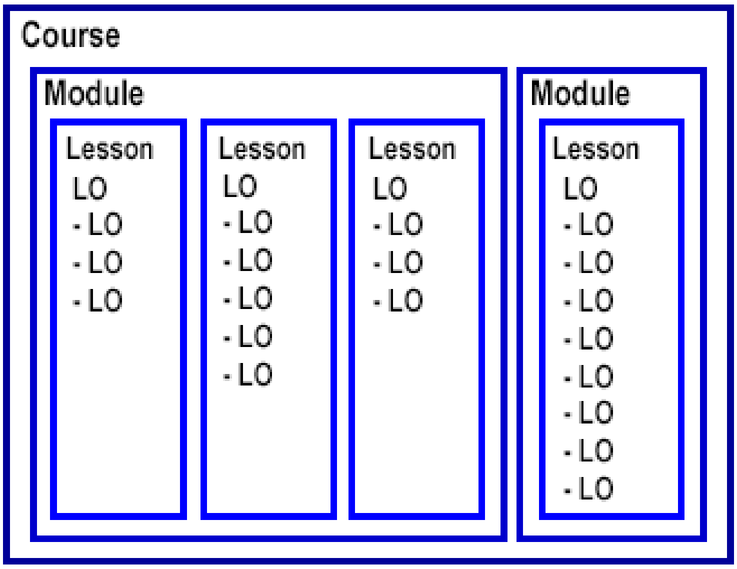
2.1.1 LO grouping

LO statements can be organized into multiple groups to optimize the management, delivery, and effectiveness of training. Instructional designers often use the following terms for LO groupings:

- a course, which is usually a series of training modules within a curriculum
- a training module, which is a stand-alone unit of training within a course that is usually comprised of multiple lessons
- a lesson, which is a segment of instruction within a training module that can often be further divided into smaller sections or topics

LO statements are commonly grouped into a course within a curriculum. A course could then have additional LO groupings that define its elements (eg, training modules and lessons) to create more granular groupings, as shown in Fig 1.

The type, scope and complexity of the Knowledge, Skill, or Attitude (KSA) to be trained for an LO can influence how it is grouped with other LO statements. For example, all LO statements in a course that address knowledge of the Product's characteristics might be grouped into a single training module, and then further grouped within that course element to address general knowledge, functional knowledge, and operational knowledge in three distinct lessons. A second training module might then be used to group all skill-based LO statements, such as the performance of preventive maintenance tasks, into a separate course element.

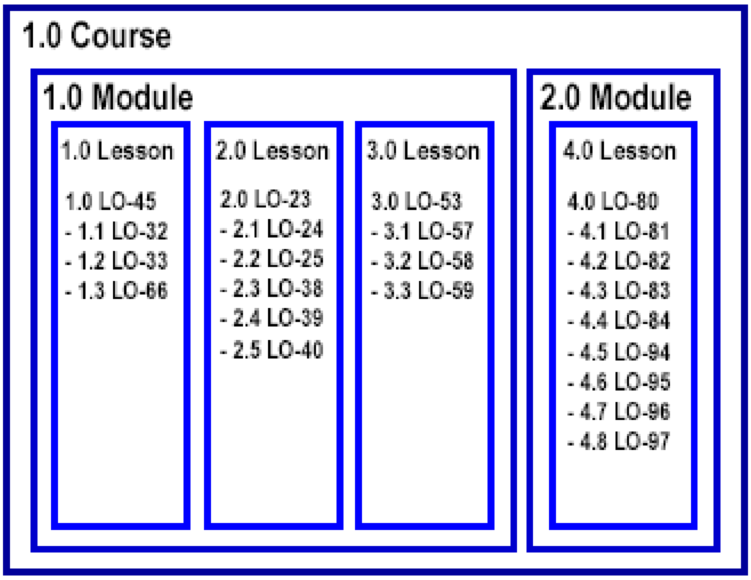


ICN-S6000T-B6865-00075-001-01

Fig 1 Example of LO statements grouped into course elements

2.1.2 LO ordering

Grouping LO statements is one aspect of a fully sequenced curriculum. The other is ordering LO groups and individual LO statements within that structure. For example, as implied by the numbering seen in Fig 2, the order of training module and lesson LO groups has been defined. The order in which individual LO statements will be trained for each lesson group has also been defined. The identifier for each LO (eg, LO-45) is shown to illustrate that this process results in LO sequencing numbers that are different from the LO identification numbers assigned during the learning objectives process (refer to Chap 4.4). Furthermore, a sequencing number for an LO could also be different in variations of the curriculum for different target audiences or changed during training updates that revise the curriculum.



ICN-S6000T-B6865-00076-001-01

Fig 2 Example of ordered LO groups and LO statements

When defining the order of LO statements within a curriculum, instructional designers commonly define terminal and enabling LO statements. In general, a terminal LO is the highest in a hierarchy of LO statements. A terminal LO will likely have supporting LO statements, which are referred to as enabling LO statements, and address the component KSA necessary to fully achieve the terminal LO.

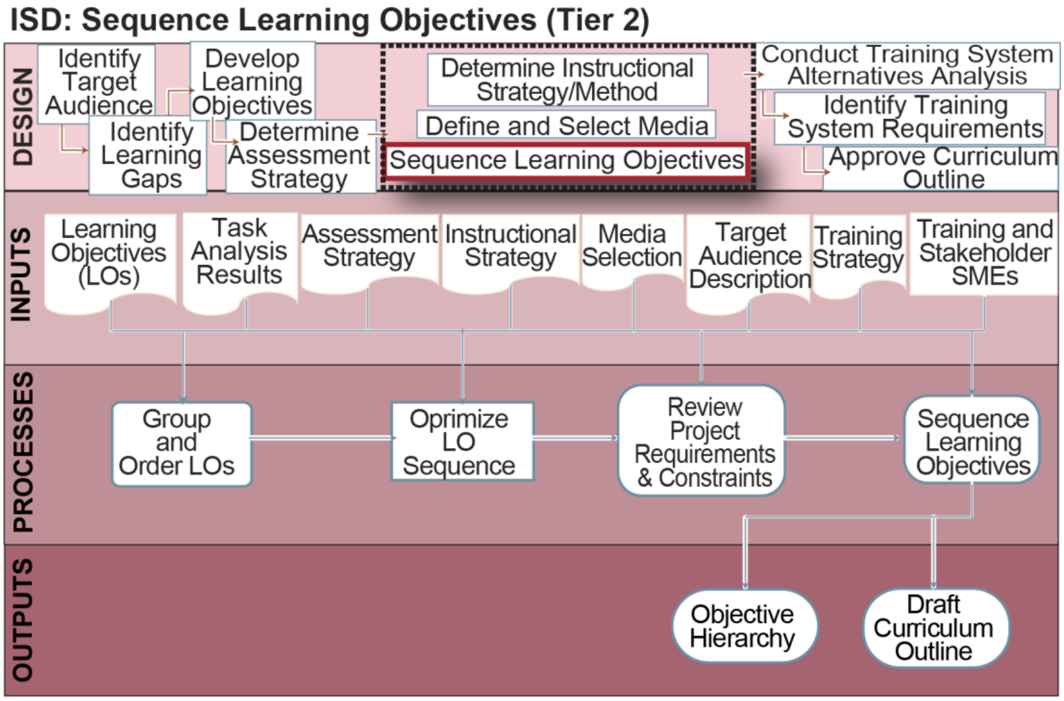
Table 2 gives an example of an LO and its supporting LO statements which are defined as terminal and enabling LO within a lesson context.

Table 2 Example of LO statements described as terminal and enabling LO statements

Lesson	**Terminal LO**	**Enabling LO**
Bicycle Corrective Maintenance	Perform basic corrective maintenance procedures on bicycles.	Repair a punctured bicycle tire. Repair a broken bicycle chain. Replace a damaged wheel. Replace damaged handlebars. Replace worn brake pads.

3 Process

Fig 3 shows the learning objective sequencing process.



ICN-S6000T-B6865-00022-002-01

Fig 3 Learning objective sequencing process

3.1 Inputs

The inputs for the learning objective sequencing process are the:

- LO, refer to Chap 4.4
- task analysis results, refer to Chap 3.5
- assessment strategy, refer to Chap 4.5
- instructional strategy, refer to Chap 4.6
- media selection results, refer to Chap 4.7
- target audience description, refer to Chap 4.2
- training strategy, refer to Chap 3.6
- training and stakeholder Subject Matter Experts (SME), refer to Para 3.1.1

3.1.1 Training and stakeholder subject matter experts

At a minimum, training and stakeholder SME who are knowledgeable in the job being trained should review the results of the learning objective sequence process. A SME can be, but is not limited to, an instructor, experienced operator and maintainer, training manager, or learning standards officer. A SME can give practical expertise that will help in effectively sequencing LO statements within and across courses and course elements. A SME should also approve the final output of this process.

3.2 Process steps

The learning objective sequencing process steps are:

- group and order LO statements, refer to Para 3.2.1
- optimize LO sequence, refer to Para 3.2.2
- review project requirements and constraints, refer to Para 3.2.3
- approve LO sequence, refer to Para 3.2.4

3.2.1 Group and order LO statements

The first step in this process is the initial grouping and ordering of all LO statements output from the learning objectives process (refer to Chap 4.4). Key considerations during this step include the:

- learning domain and level of learning for each LO, refer to Para 3.2.1.1
- relationships between LO statements, refer to Para 3.2.1.2
- training methods and media planned for each LO, refer to Para 3.2.1.3
- sequencing approach to be applied, refer to Para 3.2.1.4

3.2.1.1 Learning domain and level of learning for each LO

The learning domain and level of learning for each LO should be considered to identify the type (eg, fact learning versus critical thinking) and complexity (eg, state facts versus develop a novel solution) of training that it requires. This information assists in making a variety of decisions, depending on the sequencing approach(es) employed. It will also assist in making initial time estimates.

Table 3 gives a suggested method for applying a code to the identified type and complexity of an LO. For example, if an LO that requires the trainee to follow a procedure, it can be coded as a skill at the guided response level (ie, S.3). If an LO requires the trainee to recite a procedure, it can be coded as knowledge at the knowledge/remember level (ie, K.1).

Table 3 Suggested LO training type and complexity coding method

K-Knowledge) (Cognitive Domain	S-Skill (Psychomotor Domain)	A-Attitude (Affective Domain)
K.1 Knowledge/Remember	S.1 Perceive	A.1 Receive
K.2 Comprehend/Understand	S.2 Set	A.2 Respond
K.3 Apply	S.3 Guided Response	A.3 Value
K.4 Analyze	S.4 Mechanism	A.4 Organize
K.5 Evaluate	S.5 Complex Overt Response	A.5 Internalize/Characterize
K.6 Create/Synthesize	S.6 Origination	

3.2.1.2 Relationships between LO statements

Another important consideration in preparation for grouping and ordering LO statements is the relationships between them. LO statements will likely have the following types of relationships:

- dependent, which is an LO that cannot be achieved before another LO is achieved
- supportive, which is an LO that transfers to another LO, making it easier to achieve
- independent, which is a standalone LO that is neither dependent on nor supportive of any other LO

During the learning objectives process (refer to Chap 4.4), supporting LO statements could have been identified for a given LO. This information is useful because such LO statements likely have dependent relationships and can be identified as such. However, supporting LO statements should also be analyzed to determine those that are similar and, therefore, have a potentially supportive relationship. For example, multiple LO statements could have “remove lug nuts” in their supporting LO statements. In the final sequence, the first instance of that supporting LO could become a supportive LO for the following instances. LO statements with supportive relationships can improve both the efficiency and effectiveness of training (eg, by activating previously learned knowledge).

NOTE

Supporting LO statements defined during the learning objective process (refer to Chap 4.4) should not be confused or considered synonymous with LO statements that have supportive relationships.

Consideration of the relationships between LO statements can be helpful in determining the order of training modules and lessons. For example, if a group of LO statements for operating a console is dependent on a group of LO statements addressing knowledge of controls and indicators, then the training modules or lessons can be effectively ordered.

3.2.1.3 Training methods and media planned for each LO

The instructional strategy (refer to Chap 4.6) and media selection (refer to Chap 4.7) processes identified training methods (eg, self-paced training) and media (eg, technical training equipment (TTE) in a lab) for each LO. This information should be considered when grouping and ordering LO statements to ensure efficient use of training time and resources. For example, it can be inefficient to group LO statements that will be trained through lab practice using TTE into multiple lessons spread across different training modules. If feasible, it can be more efficient to group all the LO statements into a single lesson (eg, as a final activity at the end of a training module).

3.2.1.4 Sequencing approach to be applied

A final consideration prior to staring the initial grouping and ordering of LO statements is the sequencing approach, or combination of approaches, to use. Common LO sequencing approaches include, but are not limited to:

- job performance order, where the sequence is the order in which the trainees' duties, tasks or steps are performed in the job context, (eg, workflow defined by maintenance procedures)
- chronological order, where the sequence flows based on the order in which events occur in time (eg, system initialization events)
- criticality, where LO statements with higher levels of importance might be placed as early within the sequence as possible (eg, knowledge of personal safety hazards and potential equipment damage)
- comparative, where LO statements familiar to the trainees are sequenced before unfamiliar LO statements (eg, operating radar in open water before operation in littoral zones)
- simple to complex, where the sequence is determined based on the difficulty of LO statements (eg, basic maintenance of a product before manual troubleshooting techniques)
- job timing, where LO statements are grouped and ordered so that the sequence is timed in relation to the time at which trainees can use or can apply the KSAs on the job (eg, deferring advanced skills training until 100 flight hours is reached, or immediately before a first tour of duty)

Applying the selected sequencing approach(es) in combination with information about each LO statement's training type, complexity, relationship to other LO statements, training method and media, etc, LO statements are grouped and ordered to define an initial sequence. Instructional designers can perform the grouping and ordering activities in whichever way suits their project, customer, or professional practice.

This step concludes with a numbered set of courses (ie, one or more), course elements (eg, training modules and lessons), and individual LO statements. The numbering of course elements will define the overall chronological and hierarchical sequence of LO groups within the course, and individual LO statements that comprise the course elements, including an estimated training time defined in weeks, days, hours, or minutes.

3.2.2 Optimize LO sequence

Once the initial sequence of LO statements is defined, it should be reviewed against assessment requirements and the target audience's characteristics. This step will help to optimize the groupings and order of LO statements.

The components and timing for testing LO statements identified during the assessment strategy process (refer to Chap 4.5) should be reviewed. At a minimum, the instructional designer should review the testing type that is assigned to each LO, or grouping of LO statements, to ensure the requirements can be supported. For example, if a set of LO statements requires a pretest prior to training, then the instructional designer should ensure that all those LO statements are grouped into the same lesson.

The assessment category and assessment criticality of each LO should also be reviewed. This information can give insight that helps refine the groupings or order of LO statements. For example, LO statements identified as having a high assessment criticality level could have been overlooked during the initial sequencing, which can result in them being moved to a more optimal location within a course.

The target audience description process (refer to Chap 4.2) described the trainees' occupational experiences, qualifications, aptitudes, and existing KSA. This information should be used to refine the groupings and order of LO statements to ensure the most effective sequence is defined for the given target audience. For example, the target audience could have a higher level of practical work experience using diagnostic techniques than novices in a trade. This can influence the placement of similar knowledge-based LO statements. Instead of grouping the LO statements for new (but similar) techniques into a knowledge-based lesson, it could be optimal to group them into a skills-based lesson for a more experienced target audience, and then sequence them immediately prior to the dependent skill-based LO statements.

3.2.3 Review project requirements and constraints

The training strategy process (refer to Chap 3.6) identified program-level training budget, schedule, and resources requirements. It also identified logistical expectations, such as trainee throughput, training facility capabilities (eg, electronic classrooms, equipment labs) and availability. The information output from that process should be reviewed to ensure external factors will not adversely affect the sequence in which LO statements are trained. Refinements to the groupings and order of LO statements could result.

3.2.4 Approve LO sequence

The groupings and order of LO statements within courses and across course elements should be reviewed and approved by training and stakeholder SME. A SME can be, but is not limited to, an instructor, experienced operator or maintainer, training manager, or learning standards officer.

This final review can result in revisions to the LO statements' groupings and/or order within (or across) those groupings. It can also result in adjustments in training time estimates. Steps in this process can be revisited accordingly, as necessary.

Once the instructional designer has training and stakeholder SME approval of the overall sequence of the LO statements, the learning objective sequencing process is complete.

3.3 Outputs

The outputs of the learning objective sequencing process include a curriculum outline that defines the sequence in which groups of and individual LO statements will be addressed throughout the training for a specific target audience, including estimated training times. An objective hierarchy is also an output of this process, which defines the relationships of all LO statements.

Chapter 4.9

Design - Training system alternatives

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.6	Analysis - Training strategy
Chap 4.2	Design - Target audience description
Chap 4.6	Design - Instructional strategy
Chap 4.7	Design - Media selection
Chap 4.8	Design - Learning objective sequencing

1 General

The training system alternatives process defines optional ways to enable the training system's design based on program constraints. It considers the resources necessary for various options in relation to a curriculum outline's defined courses and course elements, the identified training methods and media, and the Learning Objective (LO) sequencing across the overall training delivery timeline. This process results in alternatives for achieving training outcomes in cases where resources are a concern or risk to successful training development and delivery, in either the near term or throughout the life cycle of a Product.

2 Description

Throughout the analysis and design phases, known constraints (eg, schedule and cost) are frequently reviewed while optimizing the training system's design. The optimized design is collectively described through the primary training method and media, LO groupings into courses and course elements, and chronological order defined for each LO within a curriculum outline. The alternatives described for the curriculum design must be based on program constraints. The description should focus on the categories of training system resources that commonly drive the need for an alternative in the development and delivery of the training system.

Training system resources are the people and assets required to develop, implement, evaluate, and maintain training systems. The major categories of resources considered throughout this process include:

- equipment/materials (eg, computers, tools, parts)
- infrastructure/facilities (eg, classrooms, training labs, engineering development labs)
- personnel (eg, instructors, trainees, technical support, engineers, lab assistants)
- documents (eg, instructor guide, trainee guide, job sheets, job aids)
- training aids (eg, video, virtual simulation, diagram, presentation, 3D model)

Training system resources have schedule, budget, and logistical requirements associated with them. Concerns or risks are identified when resource availability conflicts with program constraints. That condition is what prompts the need to define alternatives. For example, conflicts can arise when the:

- development cycle time required for a virtual simulation extends beyond the date on which it is needed in the program training schedule
- cost to modernize a building for classroom training is unallocated in the current year's program training budget
- qualified instructors needed to deliver training are already assigned elsewhere to support another program objective

2.1 Describing an alternative

The detail in an alternative's description can vary based on program and customer needs. It can also be influenced by the information that is available to an instructional systems designer (eg, labor costs). At a minimum, the alternative's description must include information about the:

- current primary method selected (refer to Chap 4.6), and the alternative method
- current primary media selected (refer to Chap 4.7), and the alternative media
- program constraint(s) it addresses
- application (ie, what course and course element it affects)
- resource differences
- tradeoffs

A change in training method or medium (or both) must be identified in an alternative's description. This information is essential for later identifying the differences in resources caused by the alternative.

An alternative's description must also identify the program constraint it addresses, and where within the curriculum outline the alternative applies. This information clarifies the purpose of the alternative by identifying the concern or risk that triggered the need for an alternative and maps it to the training system's design.

Depending on the method and/or media alternative identified, the anticipated resources can be unchanged. However, additional resources can be required. The alternative's description must identify any differences in resources, which is necessary for identifying tradeoff considerations afforded by the alternative.

When changing the design of a curriculum outline to address a concern or risk, there can be effects that should be considered before accepting or rejecting an alternative. An alternative's description must include information about tradeoffs in terms of training system efficiency, effectiveness, schedule, cost, and logistics.

Alternative descriptions are the main output of this process. They are then approved or rejected for inclusion in the training system and adjust the plan of action for developing and delivering the training curriculum outline's design.

2.2 Example alternative description

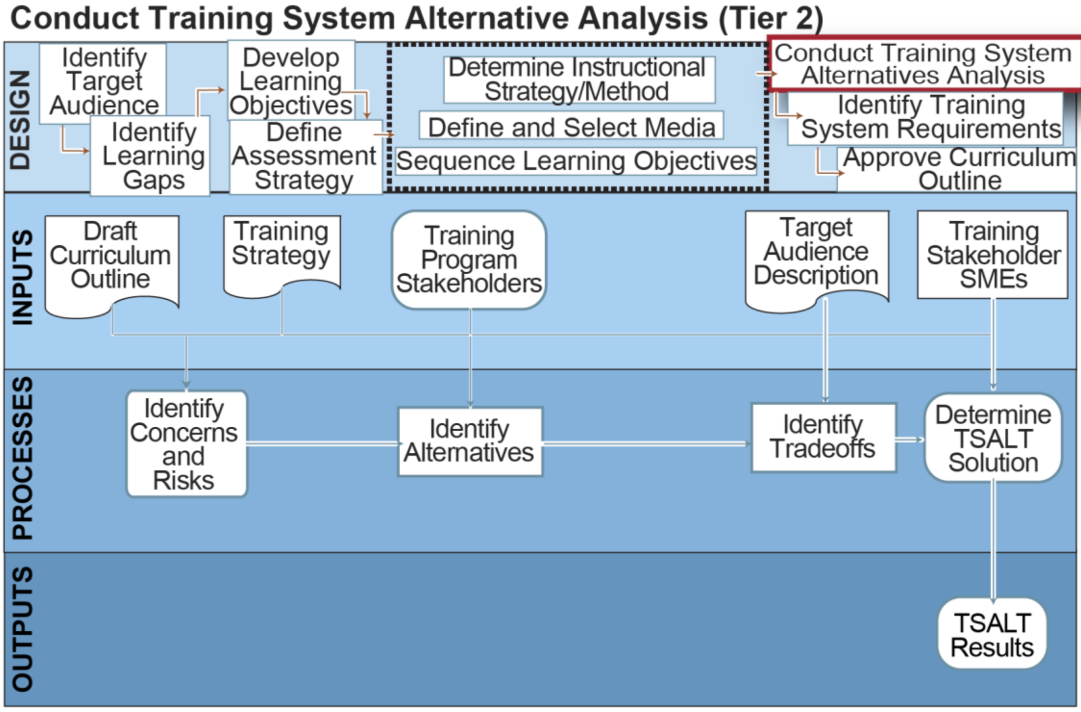
The type of information that could be contained in an alternative's description to address a schedule risk is given below. The preferred design identifies the use of simulated equipment by trainees, guided by an instructor using a facilitated instruction training method. Program schedule constraints triggered the need to define an alternative medium due to a development time risk. The description identifies a physical equipment asset as the alternative's medium and then describes the resource differences and tradeoff considerations that result.

Alternative for Risk 0001

- Current method: Facilitated instruction
- Method alternative: Not applicable. No method change is proposed
- Current media: Virtual simulated equipment
- Media alternative: Physical equipment
- Program constraints addressed
  - Schedule: Meets Q1 20xx training start date
  - Budget: N/A
- Application:
  - Course: A-NNN-NNNN
  - Course element: Module 2.0
- Resource Differences
  - Equipment/materials: System X
  - Infrastructure/facilities: Engineering Lab in location B
  - Personnel: System engineer (x2)
  - Documents: No change. Same document resources are required
  - Training aids: No change. Same training aid resources are required
- Tradeoffs
  - Training efficiency: 50% lower trainee throughput
  - Training effectiveness: No significant change
  - Program cost: 30% increase in funding requirement
  - Program schedule: Increase of 3 days in training time duration
  - Program logistics: Reservation of lab time is required but cannot be guaranteed

3 Process

Fig 1 shows the training system alternatives process.



ICN-S6000T-B6865-00023-002-01

Fig 1 Training system alternatives process

3.1 Inputs

The inputs for the training system alternatives process are:

- the draft curriculum outline description, refer to Chap 4.8
- the training strategy, refer to Chap 3.6
- training program stakeholders, refer to Para 3.1.1
- the target audience description, refer to Chap 4.2
- training and stakeholder Subject Matter Experts (SME) , refer to Para 3.1.2

3.1.1 Training program stakeholders

Input from stakeholders at the training program level is critical when identifying concerns or risks that trigger the need to identify alternatives. A part of their input is current program cost and schedule information, which can influence the severity of a concern or risk, or possibly eliminate it. Program-level stakeholders' review of alternatives is also a critical input to verifying whether a a given alternative is viable from a training program perspective.

3.1.2 Training and stakeholder SME

A training SME (eg, curriculum designer, instructor, prior trainees) should have:

- experience in the delivery of Product maintenance and operation training
- technical expertise related to the Product
- an understanding of the KSA trainees need to acquire

Training SME input is critical when considering changes in training efficiency and effectiveness an alternative can cause for the given target audience.

A stakeholder SME (eg, training manager) should have expertise in the delivery and sustainment of training systems, and a broad understanding of training program priorities. Stakeholder SME input is critical when considering how an alternative can affect training efficiency, as well as the cost, schedule, and logistical impacts of changes to resource requirements.

3.2 Process steps

The training system alternatives process steps are to:

- identify concerns and risks, refer to Para 3.2.1
- identify alternatives, refer to Para 3.2.2
- Identify tradeoffs, refer to Para 3.2.3
- define training system alternatives solution, refer to Para 3.2.4

3.2.1 Identify concerns and risks

The first step in this process is to determine whether the curriculum outline's current intended design presents any concerns or risks to success. The draft curriculum outline description resulting from the learning objectives sequencing process (refer to Chap 4.8) must be used as the baseline when identifying areas in the design that require alternatives. The exact approach depends on the scope and complexity of the training system, and how the courses and course elements are defined and sequenced in the curriculum. For example, an alternative training medium could apply to a single lesson, whereas an alternative method/media pair could be identified for an entire course.

The instructional systems designer must review the curriculum outline to find potential areas of concern that can result in program cost or schedule risks that could threaten successful development and delivery of the training system. The instructional systems designer should review, at a minimum, the:

- resource requirements described for the methods and media selected during the instructional strategy (refer to Chap 4.6) and media selection processes (refer to Chap 4.7)
- training system schedule information from the training strategy process (refer to Chap 3.6)
- program funding/budget information gathered during the training strategy process (refer to Chap 3.6)

The instructional systems designer must identify each area of concern in relation to the curriculum outline's structure. For example, computer equipment must be acquired for a course element (eg, a lesson) that will use a virtual simulation. However, the acquisition timeline needed to purchase the equipment will place its delivery date past the date on which it is needed for training. This schedule concern would be mapped to the course element.

Multiple concerns can be mapped to the same element within the curriculum outline. Building on the above example, if the computer equipment is expected to be significantly more expensive than the common consumer equipment that was initially budgeted, then this second cost-related concern could be mapped to the same lesson course element.

The instructional systems designer should then review the identified concerns with a training stakeholder SME. Their input will help qualify whether a concern remains a noted concern or escalates to become a risk that needs an alternative. For example, a training manager could know how to expedite the purchasing process for equipment. Therefore, the concern of having computer equipment in time for training can remain a concern, and no alternative needs to be defined.

Training program stakeholders are another source of similar information and can provide the deciding factor on whether alternatives need to be defined. The instructional systems designer must review all identified concerns with training program stakeholders (eg, program manager, Product engineering manager), as appropriate. Once this review is complete, alternatives are described for concerns that become escalated to risks.

3.2.2 Identify alternatives

Once risks are identified and mapped to courses and course elements in the draft curriculum outline, viable methods, or media (or both) must be identified in this step to address each risk. This is the first part of the information needed in an alternative's description.

The draft curriculum outline description represents the preferred training system design. The instructional systems designer should review information about the primary methods and media, as well as any alternates that were considered during the instructional strategy (refer to Chap 4.6) and media selection processes (refer to Chap 4.7). The instructional systems designer should also review the defined media pool for viable alternatives.

The instructional systems designer should review the identified alternatives with a training SME (eg, curriculum designer, instructor, prior trainee) who has experience in Product maintenance and operation training delivery. SME input can help in refining the resource differences for an alternate method or media.

3.2.2.1 Example of part one in an alternative's description

An example of the information included in part one of an alternative's description is given below. Information provided in parentheses is for context only and would not be included in the description. The second part of the example alternative's description is completed in the next step (refer to Para 3.2.3.1).

Alternative for Risk 0002

- Current method: Self-paced electronic training
- Method alternative: Facilitated classroom training
- Current media: Interactive Multimedia Instruction (IMI)
- Media alternative: Instructor
- Program constraints addressed
    - Schedule: Meets Q1 20xx training start date
    - Budget: N/A
- Application:
    - Course: A-NNN-NNNN
    - Course element: Module 1.0
- Resource Differences
    - Equipment/materials: System X (Physical equipment)
    - Infrastructure/facilities: No change. (Schoolhouse classroom was already planned.)
    - Personnel: Instructor (x1)
    - Documents: No change. Same document resources will be used.)
    - Training aids: PowerPoint presentation materials, Job Sheets (for trainee exercises), etc

Note

Part two of this example was omitted for brevity.

3.2.3 Identify tradeoffs

The second part of an alternative's description is the tradeoff it causes in terms of training effectiveness and efficiency. This is influenced by the differences in resources needed for the alternative, and the resources' requirements. It is also influenced by the characteristics of the method and/or media. For example, interactive multimedia delivered over the Internet can be used by more trainees at the same time, versus a live instructor who delivers the same training content in a closed, physical classroom. Therefore, the throughput of trainees becomes a tradeoff to consider as part of an alternative's description.

The instructional systems designer should review the identified alternatives with a training SME (eg, curriculum designer, instructor, and prior trainee) who has experience in the delivery of Product maintenance and operation training. SME input can help in refining the training efficiency and effectiveness tradeoffs an alternate method or media could cause for the target audience.

The instructional systems designer should also review the identified alternatives with a stakeholder SME (eg, training manager) who has expertise in the delivery and sustainment of training systems. Stakeholder SME input can help refine the program cost, schedule, and logistical tradeoffs for an alternate method or media.

The instructional systems designer must review the identified alternatives with a training program stakeholder (eg, program manager, customer). Training program stakeholder approval of alternatives is required before the training system alternatives solution can be defined in the final step of this process (refer to Para 3.2.4.)

3.2.3.1 Example of part two in an alternative's description

An example of the information included in the second part of an alternative's description is given below. Information given in parentheses is for context only and would not be included in the description. The first part of this example description was completed in the previous step (refer to Para 3.2.2.1).

Alternative for Risk 0002

- Tradeoffs
  - Training efficiency: 80% lower trainee throughput
  - Training effectiveness: No significant impact results from this change
  - Program cost: 30% decrease in year 1. 45% increase in year 2, required to develop the deferred IMI media
  - Program schedule: Increase of 6 days in training time duration
  - Program logistics: No significant impact results from this change

Note

Part one of this example was omitted for brevity.

3.2.4 Define training system alternatives solution

This final step begins after all alternatives are fully described by the instructional systems designer and approved by a training program stakeholder. The instructional systems designer must define the revised plan of action for training system development and delivery. The training system alternatives solution is the collective definition of all approved individual alternatives within the curriculum outline. It gives the rationale and audit trail for the resulting changes to the training system's design. Additionally, the curriculum outline description is updated in accordance with the approved alternative methods and/or media.

3.3 Outputs

The outputs of this process are training system alternative descriptions, which give a comprehensive solution that is approved to address training system development and delivery risks. Individual alternatives are also mapped to and updated in the curriculum outline description as a result of this process, which results in a revised curriculum outline (refer to Chap 4.8).

End of data module

Chapter 4.10

Design - Training system requirements

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.6	Analysis - Training strategy
Chap 4.6	Design - Instructional strategy
Chap 4.7	Design - Media selection
Chap 4.8	Design - Learning objective sequencing
Chap 4.9	Design - Training system alternatives

1 General

The training system requirements process describes all resource requirements for the development and delivery of the course(s) and course elements described in the curriculum outline. It identifies what is needed in terms of project requirements (ie, schedule, budget, and logistical) and functional requirements (eg, technical specifications) for each resource needed for the training system. This process results in the description of training system requirements to the level of detail needed for planning and acquiring resources, communicating requirements for training product development, and delivering the training.

2 Description

This process focuses on writing requirement statements for resource items that specify what is needed to develop, deliver, and support the training methods and media identified in the curriculum outline description.

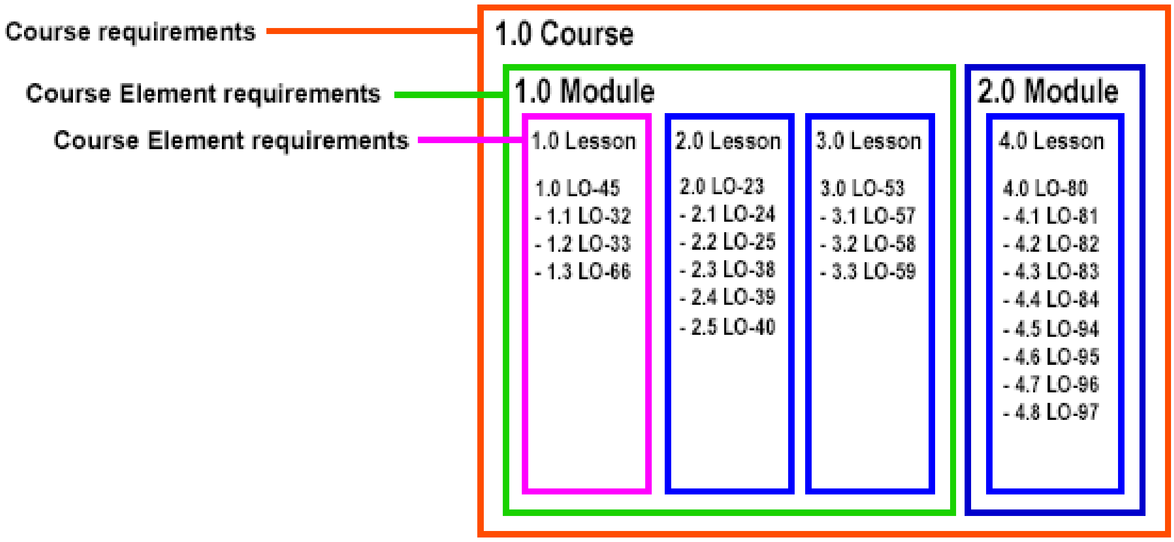
Training system requirements are described as:

- project requirements, which identify resources' schedule, budget, and logistical requirements for training system development, delivery, and support
- functional requirements, which identify resources' minimum feature, functionality, content, performance, and/or behavioral requirements in the context of a training system at completion of development and delivery

2.1 Describing training system requirements

Requirements are organized in relation to the curriculum outline resulting from the learning objective sequencing process (refer to Chap 4.8). Project and functional requirements are described for the resource item to which they apply. However, the instructional designer only describes requirements that are specific to the training system during this process. Existing requirements that are inherited from and owned by other program elements (eg, engineering, product support, maintenance planning) are not duplicated during this process. Instructional designers only define unique training-related requirements.

The approach used to map requirements to the curriculum outline will depend on how the training curriculum is designed, its complexity, and program or customer needs. For example, consider the moderately complex design shown in Fig 1. Requirements can be identified for a module or lesson, which are the more granular course elements described in the example curriculum outline. However, it can be more practical to identify requirements for some resource items (eg, computer equipment that will be used for multiple lessons) at the course level. The instructional designer can use whichever single or combined approach is most useful for managing the given requirements, but they must be associated to the curriculum outline's course and course elements.



ICN-S6000T-B6865-00077-001-01

Fig 1 Mapping resource requirements to a curriculum outline's components

The level of detail in a training system's requirements description will depend on program and customer needs, the training system's scope, and the curriculum's method and media complexity. It can also depend on the availability of information at the point in time requirements are being described.

Requirement statements are the main output of this process. They are then approved or rejected for inclusion in the training system requirements description.

2.2 Requirement statement examples

Table 2 thru Table 6 give examples of requirement statements that could be contained in a training system requirements description. It uses an approach that relates the identified resource items' requirements to a training curriculum's lesson, which is a course element. Other approaches can be used or combined, as needed.

Table 2 Equipment/material requirements

Equipment/material requirements	
Required item	System equipment X
Budget	No additional training requirements.
Schedule	Training configuration and test duration shall be 5 days.
Logistics	Equipment use shall be coordinated with system engineering.
Functional	– System shall simulate fault condition A. – System shall simulate fault condition B. – System shall… etc.

Table 3 Infrastructure/facility requirements

Infrastructure/facility requirements	
Required item	Engineering lab in location B.

Infrastructure/facility requirements	
Budget	No additional training requirements.
Schedule	Trainees shall have access for 20 days.
Logistics	Lab use shall be coordinated with system engineering.
Functional	No additional training requirements.

Table 4 Personnel requirements

Personnel requirements	
Required item	System engineer
Budget	Training program shall provide budget of $nn.
Schedule	System engineer time shall equate to 1.25 FTE over 20 days.
Logistics	No additional training requirements.
Functional	No additional training requirements.
Required item	Instructor
Budget	Training program shall provide budget of $nn.
Schedule	Instructor time shall equate to 1.0 FTE over 20-day duration.
Logistics	No additional training requirements.
Functional	No additional training requirements.

Table 5 Documents requirements

Documents requirements	
Required item	System X maintenance and troubleshooting manual
Budget	No additional training requirements.
Schedule	Manual shall be complete 15 days prior to training start date.
Logistics	Manual shall be distributed 10 days prior to training start date.
Functional	No additional training requirements.

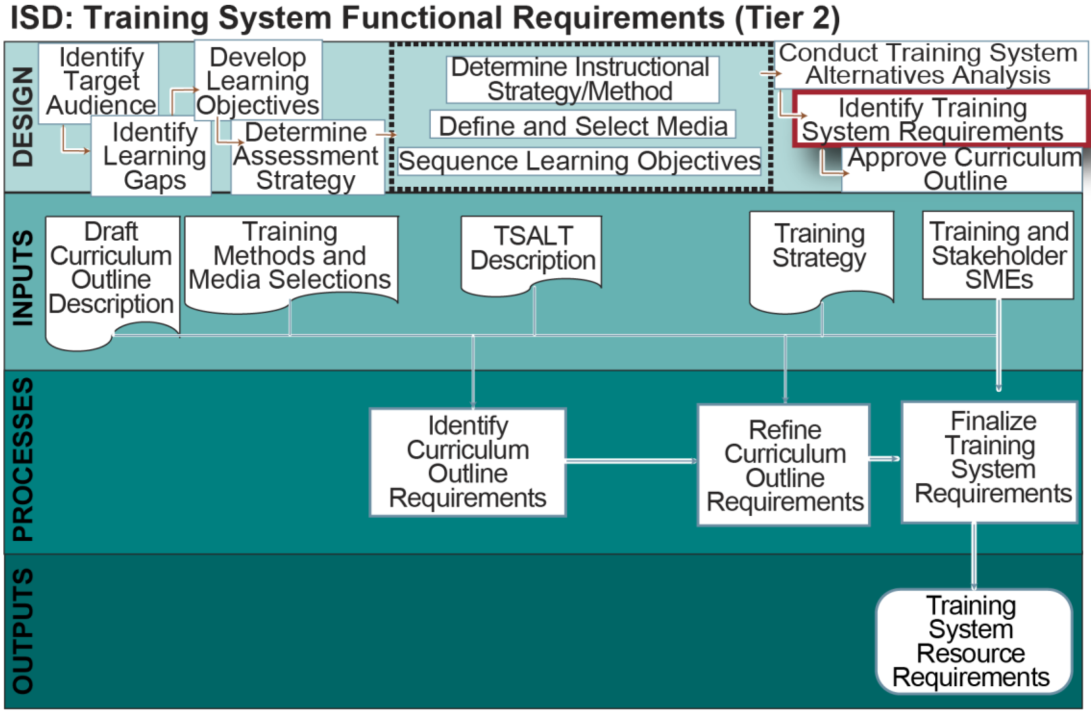
Table 6 Training aids requirements

Training aids requirements	
Required item	System X signal flow animation
Budget	Training program shall include animations in the media budget.
Schedule	Item shall be complete 30 days prior to training start date.
Logistics	Item shall be included in training program's media package.

Training aids requirements	
Functional	– Animation shall show signal flow under normal conditions. – Animation shall show signal flow under fault condition A. – Animation shall show signal flow under fault condition B. – Animation shall etc

3 Process

Fig 2 shows the training system requirements process.



ICN-S6000T-B6865-00024-002-01

Fig 2 Training system requirements process

3.1 Inputs

The inputs for the training system requirements process are:

- draft curriculum outline description, refer to Chap 4.8
- training method selections, refer to Chap 4.6
- media selections, refer to Chap 4.7
- training system alternatives description, refer to Chap 4.9
- training strategy, refer to Chap 3.6
- training and stakeholder Subject Matter Experts (SME), refer to Para 3.1.1

3.1.1 Training and stakeholder subject matter experts

A training SME (eg, curriculum designer, instructor, prior trainees) should have:

- experience in product maintenance and operation training delivery
- technical expertise related to the Product
- an understanding of the KSA trainees need to acquire

Training SME input is critical to ensure that the resources needed for training production teams and delivery personnel to successfully develop and deliver the design are completely described.

A stakeholder SME (eg, training manager) should have expertise in the delivery and sustainment of training systems, and a broad understanding of training program execution. Stakeholder SME input is critical to ensuring that the resource requirements for training production teams and personnel to successfully develop and deliver the design are completely identified. Their input can also be helpful when estimating the training project budget, schedule, and logistics requirements caused by required resources.

3.2 Process step

The training system requirements process steps are:

- identify curriculum outline requirements, refer to Para 3.2.1
- refine curriculum outline requirements, refer to Para 3.2.2
- finalize training system requirements, refer to Para 3.2.3

3.2.1 Identify curriculum outline requirements

The curriculum outline description resulting from the learning objectives sequencing process (refer to Chap 4.8) must be used as the point of reference when identifying resource requirements. The instructional designer must decide on an approach for mapping requirements to the curriculum outline's structural components. A singular or combined approach can be used, as needed. Refer to Para 2.2 for an example approach.

Project requirements (ie, budget, schedule, logistics) and functional requirements must be identified during this step. Multiple project-level factors can influence which requirements need to be defined, and what amount of detail is needed. It is out of the scope of this process, and impractical, to account for the complexity of all such factors when deriving training resource requirements. Therefore, this process gives general guidance on the use of outputs from previous processes and the types of considerations that should be made to identify requirements.

At the completion of this step, the instructional designer must have an initial set of project and functional requirements identified for each resource item. The instructional designer must also ensure that all requirements are specific to the training system. If applicable, duplicates of existing requirements that are inherited from and owned by other program elements must be removed from the training curriculum's resource requirement statements.

3.2.1.1 Project requirements

The instructional designer must identify the project requirements for each resource. Project resources are defined as:

- budget requirements, which define what it will cost to develop, acquire, deliver, or use the resource
- schedule requirements, which define how much time is needed for, or the timing of, the development, acquisition, delivery, or use the resource
- logistics requirements, which define any dependencies or special considerations required to coordinate the development, acquisition, delivery, or use of the resource

Most of resource requirements are driven by the training methods and media planned into the course and course elements' design. Training methods identified in the curriculum outline description resulted from the instructional strategy process (refer to Chap 4.6), and media from the media selection process (refer to Chap 4.7). Alternative training methods or media could also have been identified during the training system alternatives process (refer to Chap 4.9). All those processes also took into consideration schedule and budget constraints defined for the training system during the training strategy process (refer to Chap 3.6). The instructional designer must take the information from all these processes in consideration when determining resource items' project requirements. For example, the training method/media descriptions related to a desktop simulation can give insight on the computer hardware needed to run the simulation in an electronic classroom. The instructional designer can use this information with training budget and schedule information to identify that equipment resource item's project requirements, which could include:

- Budget: - Training program shall provide budget of $nn
- Schedule: - Computer setup shall be completed 30 days prior to training event

3.2.1.2 Functional requirements

The instructional designer must also identify the functional requirements for each resource. Functional requirements define the technical aspects of resources. For example, if a simulation must support the performance of maintenance procedures, then a functional requirement could be identified for each procedure. Furthermore, if the simulation must run on existing computer equipment that has known specifications (eg, 8 gigabytes of memory), then appropriate functional requirements should be identified for that simulation resource.

3.2.2 Refine curriculum outline requirements

Requirements must be reviewed by a training SME and stakeholder SME to ensure that they are accurate, complete, and give the level of detail needed for training development and delivery. A training SME (eg, curriculum designer, instructor, training software developer) generally review functional requirements to confirm adequacy for training production teams and personnel to successfully develop and deliver the curriculum outline's planned design. A stakeholder SME (eg, training project manager) will generally confirm that project-oriented cost, schedule, and logistical requirements are accurate and complete. However, any SME can have valuable insight that assists in refining project requirements and functional requirements.

A stakeholder SME should also provide input on how functional requirements will be verified. While refining requirements, the instructional designer should identify how each functional requirement will be verified. Requirements verification can be accomplished through:

- analysis, which uses recognized analytic techniques (including computer models) to interpret or explain the behavior/performance of the system element. Analysis of test data or review and analysis of design data should be used to verify requirements, as appropriate.
- inspection/examination, which uses visual inspection to verify conformance with overt characteristics, such as physical, material, part and product marking and workmanship.
- test, which uses an activity designed to give data on functional features and operation under fully controlled and traceable conditions. The data are subsequently used to verify quantitative characteristics.
- demonstration, which uses the performance of a product, where visual observations are the primary means of verification. Demonstration is used when quantitative measures are not required for verification.

3.2.3 Finalize training system requirements

The final step in this process begins after all requirements are reviewed by a training SME and stakeholder SME. The instructional designer must revise the set of requirements in accordance with the collected SME input, including verification methods, if applicable. Once the revisions to requirement statements are complete, the training system requirements are finalized.

3.3 Outputs

The output of this process is the training system resource requirements description, which is the complete set of project requirement and functional requirement statements for the curriculum outline.

Chapter 4.11

Design - Curriculum outline approval and release

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 4.6	Design - Instructional strategy
Chap 4.7	Design - Media selection
Chap 4.8	Design - Learning objectives sequencing
Chap 4.10	Design - Training system requirements

1 General

The curriculum outline approval and release process, places the training curriculum outline under version control and formalizes it as part of the training system. It includes the approval of the defined instructional design and resources in the curriculum outline (refer to Chap 4.8) as well as the training system requirements (refer to Chap 4.10). The curriculum outline approval and release process results in the release of a customer-approved curriculum outline for planning and acquiring resources and communicating resource requirements for training product development and delivery.

2 Description

This process gives guidance for finalizing the curriculum outline's content, structure, and associated resource requirements. It focuses on approval of the curriculum outline's design in parallel with the project (ie, schedule, budget, and logistics) and functional requirements for training development and delivery.

Approval of the curriculum outline and its resource requirements is achieved through reviews that consider whether the:

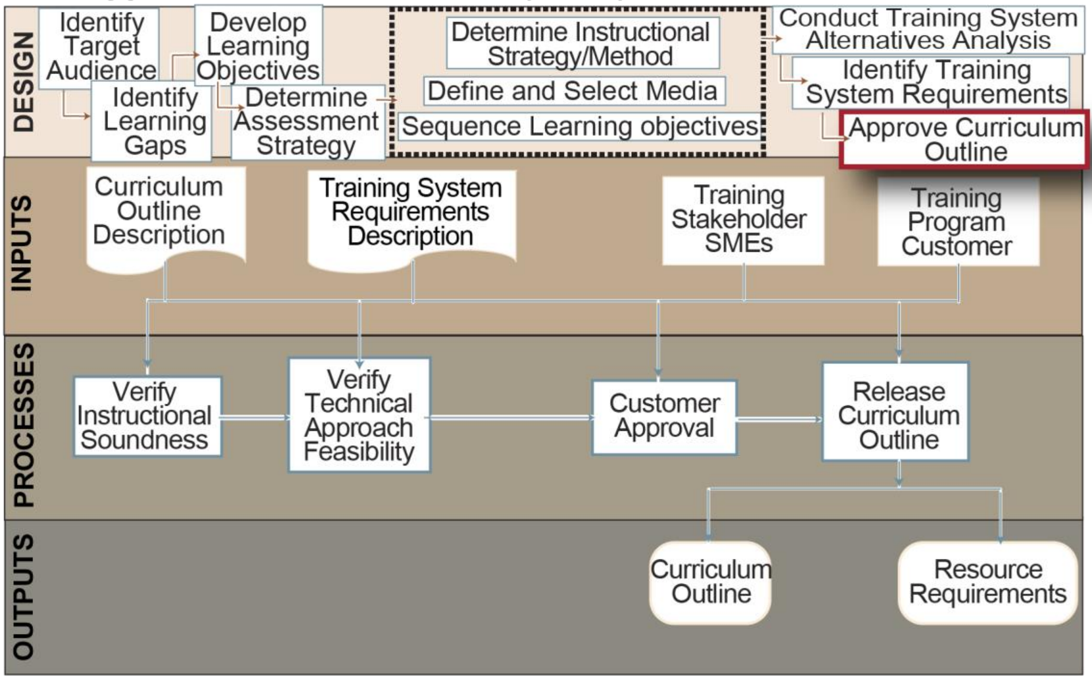
- design is instructionally sound, which verifies the inherent training efficiency and effectiveness of the curriculum outline's design
- technical approach is feasible, which verifies that training product development and delivery can be achieved with the project and functional requirements identified for the resources

Once reviews are complete, and any resulting revisions are satisfied, the curriculum outline description and associated resource requirements are approved by the customer. The curriculum outline is then placed under version control. For example, if the curriculum outline is for a new Product that will be trained for the first time, it would likely become version 1.0. If it is for a change to an existing Product and the training is being updated, then the version could be version 1.1.

The approved curriculum outline is then released for use in the development phase processes. For example, software designers can proceed with the development of a simulation once they have the functional requirements of the approved version released to them.

3 Process

Fig 1 shows the process to approve and release the curriculum outline.



ICN-S6000T-B6865-00025-002-01

Fig 1 Curriculum outline approval and release process

3.1 Inputs

The inputs to the curriculum outline approval and release process are the:

- curriculum outline description, refer to Chap 4.8
- training system requirements descriptions, refer to Chap 4.10
- training and stakeholder Subject Matter Experts (SME), refer to Para 3.1.1
- training program customer, refer to Para 3.1.2

3.1.1 Training and stakeholder subject matter experts

A training SME (eg, curriculum designer, instructor, and prior trainee) should have:

- experience in conducting product maintenance and operation training
- technical expertise related to the Product
- an understanding of technical training development

Training SME input helps verify that the curriculum outline's design is efficient and effective for the target audience. It also helps verify that the resource requirements are adequate to support development and delivery of the training products.

A stakeholder SME (eg, training manager) should have expertise in the delivery and sustainment of training systems. They should also have a broad understanding of training program development and execution. Stakeholder SME input helps verify that the resource requirements are adequate for training production teams and personnel to produce and deliver the curriculums outline's design.

3.1.2 Training program customer

Training program customer input is critical to the final approval of the curriculum outline and its associated resource requirements. Depending on the training project, the customer's review can be completed by one person (eg, program manager) or a group (eg, curriculum control board). The training program customer's review ensures that the curriculum outline and its resource requirements are approved and can be released for development.

3.2 Process steps

The curriculum outline approval and release process steps are:

- verify instructional soundness, refer to Para 3.2.1
- verify technical approach feasibility, refer to Para 3.2.2
- customer approval, refer to Para 3.2.3
- release curriculum outline for development, refer to Para 3.2.4

3.2.1 Verify instructional soundness

During this step, a training SME performs a comprehensive review of the curriculum outline description output from the learning objective sequencing process (refer to Chap 4.8). The design of the curriculum outline must be reviewed for training efficiency and effectiveness. For example, the training SME should verify that the sequencing of courses and course elements gives an optimal learning path for the target audience. They should also verify that the methods and media used in the course elements support the intended learning outcomes.

The instructional systems designer must collect training SME input and resolve it accordingly. This review can result in design revisions that reiterate the instructional strategy (refer to Chap 4.6), media selection (refer to Chap 4.7), or learning objective sequencing processes (refer to Chap 4.8).

3.2.2 Verify technical approach feasibility

Both a training and stakeholder SME should verify that the design can be achieved within the project and functional requirements identified for the curriculum outline's resources. During this step, each SME should perform comprehensive reviews of the resource requirements output from the training system requirements process (refer to Chap 4.10). They must review the curriculum outline for technical feasibility in terms of the identified resource requirements' adequacy to support training development and delivery. For example, if a simulation is a resource in the design, a training SME should verify that it can be developed and delivered with the identified schedule, budget, and functional requirements.

The instructional systems designer must collect SME input and resolve any comments, corrections, or revisions accordingly. This review can result in requirement changes that reiterate the training system requirements process (refer to Chap 4.6).

3.2.3 Customer approval

After SME verification that the curriculum outline is instructionally sound and feasible, and is complete, the customer must give final approval. How the customer conducts this review can vary among projects. For example, the customer's review can be completed by one person (eg, program manager) or a group (eg, curriculum control board). Furthermore, one customer could only be concerned with project-related resource requirements, such as budget and schedule. Another customer could conduct their own detailed review for instructional soundness and feasibility (or reasonableness) of the training design. The instructional designer should be aware of the customer's review criteria and support accordingly.

The customer's review can result in design revisions that reiterate the instructional strategy (refer to Chap 4.6), media selection (refer to Chap 4.7), learning objectives sequencing (refer to Chap 4.8), or training system requirements (refer to Chap 4.10), processes. Once the customer has approved the curriculum outline and its resources' requirements, the instructional designer must then be assigned an identifier to indicate the final, approved version of the curriculum outline (eg, 1.0, 1.1, 2.0, etc).

3.2.4 Release curriculum outline for development

In this final step, the final version of the curriculum outline that is approved by the customer can be released to all stakeholders and product development teams.

3.3 Outputs

The outputs of this process are the customer-approved course curriculum outline and resource requirements that will be used for training product development and delivery. This completes the design phase, which gives the complete set of information needed to plan for and acquire resources, develop training products, and deliver the training.

Chapter 4.12

Design - Business objects

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 4.1	Design - Introduction
Chap 4.2	Design - Target audience description
Chap 4.3	Design – Learning gaps
Chap 4.4	Design - Learning objectives

Chap No./Document No.	Title
Chap 4.5	Design - Assessment strategy
Chap 4.6	Design - Instructional strategy
Chap 4.7	Design – Media selection
Chap 4.8	Design - Learning objective sequencing
Chap 4.9	Design - Training system alternatives
Chap 4.10	Design - Training system requirements
Chap 4.11	Design - Curriculum outline approval and release
Chap 7	Data model
Chap 7.3.7	Units of functionality - UoF Course
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course plan
Chap 7.3.15	Units of functionality - UoF Learning Assessment
Chap 7.3.16	Units of functionality - UoF Learning Gap
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.39	Units of functionality - UoF Training Entry Requirement
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity
Chap 7.3.42	Units of functionality - UoF Training Method
Chap 7.3.44	Units of functionality - UoF Training System Resource Requirement
SX001G	Glossary for the S-Series IPS specifications

1 General

This chapter defines the most important business objects and data elements captured during the design phase activities described in Chap 4.2 through Chap 4.11.

Business objects and data elements common to other S-Series IPS specifications are defined with a reference to the SX001G specification, which enables harmonization and interoperability across all S-Series IPS specifications.

1.1 Scope

The following types of data elements are within scope for the S6000T training design activities. Data elements that:

- identify and describes the target audience
- define gaps in Knowledge, Skills and/or Attitudes (KSA) for the target audience
- define learning objectives to address identified gaps in KSA
- identify assessment strategies and instructional strategies for the respective learning objectives
- define a curriculum for the target audience

- define courses and course elements that address the identified learning objectives
- define training system resource requirements

1.2 Out of scope

This chapter does not give all the details that are defined in the data model (refer to Chap 7) but highlights the most important business objects and data elements that need to be documented as part of the design activities.

2 Business objects and data elements

Business objects supporting the S6000T training design process are organized in accordance with the respective training design chapter. The purpose for defining business objects is to either reflect something that exists in the real world or define a placeholder for information that belongs together from a logical standpoint. Data elements then define the type of data that can be captured for the respective business object.

Each business object has a reference to the Unit of Functionality (UoF) in Chap 7 where complete definitions of its data elements (characteristics) are defined in the context of the S6000T data model.

2.1 Target audience description

The objective for the target audience description is to define and describe the characteristics of the people who will participate in the training including their KSA ratings in relation to the Product to be operated and maintained. (Refer to Chap 4.2).

2.1.1 Business objects - Target audience analysis

Business objects and data elements which are defined during mission analysis are listed in Table 2.

Table 2 Target audience analysis business objects and data elements

Term	Definition
Training entry requirement *(Business object)*	Training entry requirement defines prerequisites that must be fulfilled by a person that will be entering a training program for a job. **Reference** – Chap 7.3.39 UoF Training Entry Requirement
Occupational background *(Business object)*	Occupational background defines perceived KSA common to a set of persons within the target audience designated to perform a certain job. **Reference** – Chap 7.3.31 UoF Target Audience
Occupational background name *(Data element)*	Occupational background name is a name by which the occupational background is known and can be easily referenced.
Audience profile characteristics *(Business object)*	Audience profile characteristics define target population intrinsic characteristics. **Reference** – Chap 7.3.31 UoF Target Audience

Term	Definition
Audience general characteristics *(Business object)*	Audience general characteristics define target population extrinsic characteristics. **Reference** – Chap 7.3.31 UoF Target Audience
Occupational background KSA subject *(Business object)*	Occupational background KSA subject defines the perceived KSA in a specific field which is important to evaluate and highlight in the context of the defined job. **Reference** – Chap 7.3.31 UoF Target Audience
Attitude level rating *(Data element)*	Attitude level rating is a classification of a person's expected level of sophistication with respect to emotional, motivational and/or social abilities in the context of the defined field of experience.
Knowledge level rating *(Data element)*	Knowledge level rating is a classification of a person's expected level of sophistication with respect to mental, cognitive, and/or logical aspects in the context of the defined field of experience.
Skill level rating *(Data element)*	Skill level rating is a classification of a person's expected level of sophistication with respect to physical, mechanical, and/or movement-based aspects in the context of the defined field of experience.
Competency definition *(Business object)*	Competency definition represents measurable or observable KSA necessary for successful performance by a person in a specific context. **Reference** – Chap 7.3.6 UoF Competency Definition

2.2 Learning gaps

The objective for the learning gaps (refer to Chap 4.3) is to document trainees' current KSAs in relation to the KSA required to perform the tasks which are selected and prioritized for training (refer to Chap 3.4).

2.2.1 Business objects - Learning analysis

Business objects and data elements which are defined during learning analysis are listed in Table 3.

Table 3 Learning analysis business objects and data elements

Term	Definition
KSA gap definition *(Business object)*	KSA gap definition defines a deficiency in KSA that must be addressed for the identified target audience. **Reference** – Chap 7.3.16 UoF Learning Gap
KSA gap definition learning domain *(Data element)*	KSA gap definition learning domain is a classification that determines to which learning domain the gap is written (knowledge, skill, or attitude)

Term	Definition
KSA gap definition target level of learning (*Data element*)	KSA gap definition target level of learning is a classification that identifies the required level of sophistication.
KSA gap definition current level of learning (*Data element*)	KSA gap definition current level of learning is a classification that identifies estimated level of sophistication for the target audience at the time of entering training.
KSA gap definition scheduled readiness requirement (*Data element*)	KSA gap definition scheduled readiness requirement is a description that specifies when the target audience must have the full operational capability.
Task KSA gap *(Business object)*	Task KSA gap identifies the task context where the deficiency in qualification has been identified. **Reference** – Chap 7.3.16 UoF Learning Gap

2.3 Learning objectives

The purpose for defining learning objectives (refer to Chap 4.4) is to express the goals of a learning outcome. Each learning objective is a precise statement that defines the behavior a trainee is expected to demonstrate, the conditions under which that behavior will be performed, and the degree of measurement based on a standard of performance.

2.3.1 Business objects - Learning objectives

Business objects and data elements which are defined during learning objective analysis are listed in Table 4.

Table 4 Learning objective business objects and data elements

Term	Definition
Learning objective *(Business object)*	Learning objective is a statement that defines the expected outcome of a learning activity in terms of assessable KSA that will be acquired by a trainee as a result of instruction. **Reference** – Chap 7.3.17 UoF Learning Objective
Learning objective identifier (*Data element*)	Learning objective identifier is an identifier that establishes a unique designator for a learning objective and to differentiate it from other instances of learning objective.
Learning domain *(Data element)*	Learning domain is a classification that determines the learning aspect from which the learning objective is written (knowledge, skill, or attitude).
Learning objective entry level requirement description *(Data element)*	Learning objective entry level requirement description gives more information on the KSA that is a prerequisite for how the learning objective is written.
Learning objective complexity (*Data element*)	Learning objective complexity is a classification that identifies the span and extent of the content needed to attain the defined knowledge, skill, or attitude.

Term	Definition
Learning performance objective (*Business object*)	Learning performance objective defines the precise statement of performance expected at the end of the learning activity. **Reference** – Chap 7.3.17 UoF Learning Objective
Learning performance objective condition *(Data element)*	Learning performance objective condition is a description that expresses the circumstances under which a trainee demonstrates the defined KSA.
Learning performance objective behavior *(Data element)*	Learning performance objective behavior is a description of the action that demonstrates that a trainee has acquired the defined KSA.
Learning performance objective standard *(Data element)*	Learning performance objective standard is a statement that describes the criteria to which a trainee will be measured when demonstrating the defined KSA.
Learning objective key learning point (*Business object*)	Learning objective key learning point defines a specific detail, aspect or quality that is important in the context of the learning objective. **Reference** – Chap 7.3.17 UoF Learning Objective
Learning objective relationship *(Business object)*	Learning objective relationship is a relationship where one learning objective relates to another learning objective. **Reference** – Chap 7.3.17 UoF Learning Objective
Learning objective relationship type *(Data element)*	Learning objective relationship type is a classification that identifies the meaning of the established relationship. **Examples** – prerequisite learning objective which defines a sequential dependency – enabling learning objective which defines a hierarchical dependency

2.4 Assessment strategy

The objective for the assessment strategy (refer to Chap 4.5) is to define how to test the expected learning outcome.

2.4.1 Business objects - Assessment strategy

The most central data elements captured during the assessment strategy analysis are listed in Table 5.

Table 5 Assessment strategy business objects and data elements

Term	Definition
Learning objective assessment definition (Business object)	Learning objective assessment definition specifies requirements for measuring and asserting the target audience's attainment of the associated learning objective. **Reference** – Chap 7.3.15 UoF Learning Assessment
Learning objective assessment performance criticality rating (Data element)	Learning objective assessment performance criticality rating is a classification that defines the need for assessing the learning objective relative to how essential the defined KSA is to competent task performance.
Learning objective assessment criticality rating (Data element)	Learning objective assessment criticality rating is a classification that determines the importance of formally testing a learning objective.
Learning objective assessment category (Data element)	Learning objective assessment category is a classification that identifies the overall test approach for measuring and asserting the target audience's attainment of the learning objective. **Examples** – knowledge assessment – performance assessment
Learning objective assessment test component (Business object)	Learning objective assessment test component defines a specific evaluation and measurement of the target audience's attainment of the associated learning objective or portion thereof. **Reference** – Chap 7.3.15 UoF Learning Assessment
Learning objective assessment test type (Data element)	Learning objective assessment test type is a classification that defines the placement and purpose of a given test component within the overall assessment strategy. **Examples** – pretest – diagnostic test – summative test
Learning objective assessment test instrument (Data element)	Learning objective assessment test instrument is a classification that defines the type of instrument that will be used for measuring the target audience's attainment of the associated learning objective or portion thereof. **Examples** – multiple choice (Knowledge test) – multiple selection (Knowledge test) – checklist (Performance test) – procedure (Performance test)

Term	Definition
Learning objective assessment test method (Data element)	Learning objective assessment test method is a classification that defines the scoring method that will be used when testing achievement of the associated learning objective or portion thereof. **Examples** – criterion referenced test method – norm referenced test method

2.5 Instructional strategy

The objective of the instructional strategy (refer to Chap 4.6) is to identify the training method(s) to be used to impart the required KSA.

2.5.1 Business objects - Instructional strategy

Data elements used to document the results from the instructional strategy analysis are listed in Table 6.

Table 6 Instructional strategy business objects and data elements

Term	Definition
Learning objective training method *(Business object)*	Learning objective training method identifies a technique which is suitable to support the acquisition of new knowledge, skills, and attitudes to meet the learning objective. **Reference** – Chap 7.3.43 UoF Training Method
Learning objective training method category *(Data element)*	Learning objective training method category is a classification that identifies the technique or format for training. **Examples** – demonstration – lecture – role play
Learning objective training method learning environment *(Data element)*	Learning objective training method learning environment is a classification that defines the how training will be delivered with respect to dependencies between trainees as well as between trainees and instructors. **Examples** – synchronous – asynchronous
Learning objective training method media fidelity requirement *(Data element)*	Learning objective training method media fidelity requirement is a description that defines the real-world characteristics a training intervention must imitate or replicate for the training for the learning objective to be meaningful.

2.6 Media selection

The objective of the media analysis activities (refer to Chap 4.7) is to identify the most efficient and effective ways to deliver learning content to trainees.

2.6.1 Business objects - Media selection

Data elements used to document the results from the media analysis activity are listed in Table 7.

Table 7 Media analysis business objects and data elements

Term	Definition
Learning objective training media *(Business object)*	Learning objective training media identifies the means, form, or vehicle by which the learning objective instruction is formatted, stored, and delivered to the trainee. **Reference** – Chap 7.3.41 UoF Training Media and Fidelity
Learning objective training media category *(Data Element)*	Learning objective training media category is a classification that identifies the delivery format of learning content to trainees.
Learning objective training media physical fidelity *(Data Element)*	Learning objective training media physical fidelity is a classification that defines how closely a training intervention for the learning objective must imitate or replicate a person’s real-world passive perception of the Product through the senses of sight, hearing, smell, touch, or taste.
Learning objective training media task fidelity *(Data Element)*	Learning objective training media task fidelity is a classification that defines how closely a training intervention for the learning objective must imitate or replicate the actual actions performed within a task.
Learning objective training media behavioral fidelity *(Data Element)*	Learning objective training media behavioral fidelity is a classification that defines how closely a training intervention for the learning objective must imitate or replicate the real-world characteristics of the Product.
Learning objective training media environmental fidelity *(Data Element)*	Learning objective training media environmental fidelity is a classification that defines how closely a training intervention for the learning objective must imitate or replicate the real-world circumstances, events and/or conditions, under which the Product is used.
Learning objective training media psychological fidelity *(Data Element)*	Learning objective training media psychological fidelity is a classification that defines how closely a training intervention for the learning objective must imitate or replicate a person's real-world mental or emotional response to the conditions under which the Product will be used.
Learning objective training media haptic fidelity *(Data Element)*	Learning objective training media haptic fidelity is a classification that defines how closely a training intervention for the learning objective must imitate or replicate a person's real-world active perception of the Product through the sense of touch or bodily interaction.

2.7 Learning objective sequencing

The objective for the learning objective sequencing activity (refer to Chap 4.8) is to identify the grouping and ordering of learning objectives for a course(s) and its component course elements (eg, training modules and lessons) to define a comprehensive training curriculum for the target audience.

2.7.1 Business objects - Learning objective sequencing

Data elements used to document the results from the learning objective sequencing activity are listed in Table 8.

Table 8 Sequence learning objectives business objects and data elements

Term	Definition
Curriculum definition (Business object)	Curriculum definition defines the training path for how a particular target audience will be trained to meet the requirements for a particular job, duty, or portion thereof. **Reference** – Chap 7.3.9 UoF Curriculum and Course Plan
Curriculum definition identifier (Data Element)	Curriculum definition identifier is an identifier that establishes a unique designator for a curriculum definition and to differentiate it from other instances of curriculum definition.
Curriculum definition purpose (Data Element)	Curriculum definition purpose is a description of the learning outcome for a trainee completing the defined curriculum definition.
Curriculum definition duration (Data Element)	Curriculum definition duration is a property that specifies the amount of time required to complete training for all courses included in the curriculum definition.
Curriculum course (Business object)	Curriculum course identifies course definitions that are part of the defined curriculum definition. **Reference** – Chap 7.3.9 UoF Curriculum and Course Plan
Curriculum definition relationship (Business object)	Curriculum definition relationship is a relationship where one curriculum definition relates to another curriculum definition. **Example** – variant of **Reference** – Chap 7.3.9 UoF Curriculum and Course Plan
Course definition (Business object)	Course definition defines a combination of course elements with the objective that a trainee is to learn knowledge, skills and attitudes to meet performance requirements for a particular aspect of a job or duty. **Reference** – Chap 7.3.7 UoF Course
Course definition identifier (Data Element)	Course definition identifier is an identifier that establishes a unique designator for a course definition and to differentiate it from other instances of course definition.
Course definition purpose (Data Element)	Course definition purpose is a description that provides information about the knowledge, skills, and attitudes a trainee is intended to learn in the defined course.
Course definition duration (Data Element)	Course definition duration is a property that specifies the amount of time required to complete training for all course elements and learning objectives included in the course definition.

Term	Definition
Course definition context (Business object)	Course definition context identifies the item(s) that is in focus for the course definition. **Examples** – Product or Product variant – system or subsystem – specific part (equipment) – defined mission **Reference** – Chap 7.3.7 UoF Course
Course included course element (Business object)	Course included course element identifies course elements that are part of the defined course definition. **Reference** – Chap 7.3.7 UoF Course
Course definition relationship (Business object)	Course definition relationship is a relationship where one course definition relates to another course definition. **Example** – prerequisite course **Reference** – Chap 7.3.7 UoF Course
Course element definition (Business object)	Course element definition defines a partition of a course definition where a trainee is to learn a particular knowledge, skill and/or attitude on a particular subject. **Reference** – Chap 7.3.8 UoF Course Element
Course element definition identifier (Data Element)	Course element definition identifier is an identifier that establishes a unique designator for a course element definition and to differentiate it from other instances of course element definition.
Course element definition category (Data Element)	Course element definition category is a classification that identifies further specialization for the course element and to position the course element in relation to the overall course structure. **Examples** – course unit – course module – course lesson
Course element definition objective (Data Element)	Course element definition objective is a description that provides information about the knowledge, skills, and attitudes a trainee is intended to learn in the defined course element.
Course element definition duration (Data Element)	Course element definition duration is a property that specifies the amount of time required to complete training for all learning objectives included in the course element definition.

Term	Definition
Course element learning objective (Business Object)	Course element learning objective identifies learning objectives that will be fully or partially covered by the course element. **Reference** – Chap 7.3.8 UoF Course Element
Course element learning objective duration (Data Element)	Course element learning objective duration is a property that specifies the amount of time required to complete training for a specific learning objective.
Course element definition relationship (Business object)	Course definition relationship is a relationship where one course element definition relates to another course element definition. **Example** – alternate to course element – prerequisite course element – child course element **Reference** – Chap 7.3.8 UoF Course Element

Chapter 5

In-Service Human Performance Optimization

Chapter 5.1

In-Service Human Performance Optimization - Introduction

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 3.2	Analysis - Training situation analysis
Chap 3.3	Analysis - Mission, job, and task analysis
Chap 3.4	Analysis - Task selection
Chap 3.5	Analysis - Training task analysis
Chap 3.6	Analysis - Training strategy
Chap 3.7	Analysis - Business objects
Chap 4.2	Design - Target audience description
Chap 4.3	Design - Learning gaps
Chap 4.4	Design - Learning objectives
Chap 4.5	Design - Assessment strategy
Chap 4.6	Design - Instructional strategy

Chap No./Document No.	Title
Chap 4.7	Design - Media selection
Chap 4.8	Design - Learning objective sequencing
Chap 4.9	Design - Training system alternative
Chap 4.10	Design - Training system requirements
Chap 4.11	Design - Curriculum outline approval and release
Chap 4.12	Design - Business objects
Chap 5.2	In-service human performance optimization - Preparation phase
Chap 5.3	In-service human performance optimization - Analysis phase
Chap 5.4	In-service human performance optimization – Follow-Up phase
S5000F	International specification for in-service data feedback

1 Introduction

ISHPO is a comprehensive process that evaluates the user's human performance when operating and maintaining a Product. At various points during a Product's lifecycle, ISHPO examines the effectivity of the existing Integrated Product Support (IPS) and how it is currently affecting human performance.

During the in-service phase of a Product's lifecycle, the ISHPO process can be triggered by several factors that can provide inputs to the process, which include, but are not limited to:

- new technology becoming available
- changes to the usage and/or mission/job of the product
- in-service experience
- observations, by either or both the supplier and the customer, of areas of training support that could be improved upon
- changes in design
- changes in other IPS elements, engineering domains or S-Series IPS specifications

The concepts of In-Service Human Performance Optimization (ISHPO) included in S6000T, are based entirely a process included in S4000P that is called In-Service Maintenance Optimization (ISMO).

Note

The patent is defined as "Determined state of the art". The patent numbers are:

- DE 20 2011 004 519 U1 (German)
- US 2012/0 323 615 (United States)

2 ISHPO process

The ISHPO process comprises three phases:

- The preparation phase, which is a detailed examination of the external factors that are inputs to the ISHPO process, including confirmation that feedback received during a Product's in-service phase, is a matter for human performance improvement. Assuming that any of these factors are considered candidates for inputs to the ISHPO process, the output of this phase will identify the actions required to analyze and address any human performance issues. Refer to Chap 5.2.

- The analysis phase, which is a detailed analysis of the Product training and its associated technical publications and logistics. The output of this phase identifies those aspects of S6000T that are candidates for updates or improvements. Refer to Chap 5.3.
- The follow-up phase, which is a detailed process that guides the training analyst and designer through the various training updates and improvements that have been identified during the ISHPO analysis phase. The output of this phase is a recommendation that states whether any human performance issue has been rectified, and therefore no further ISHPO action is required, or if the issue requires further investigation. Refer to Chap 5.4.

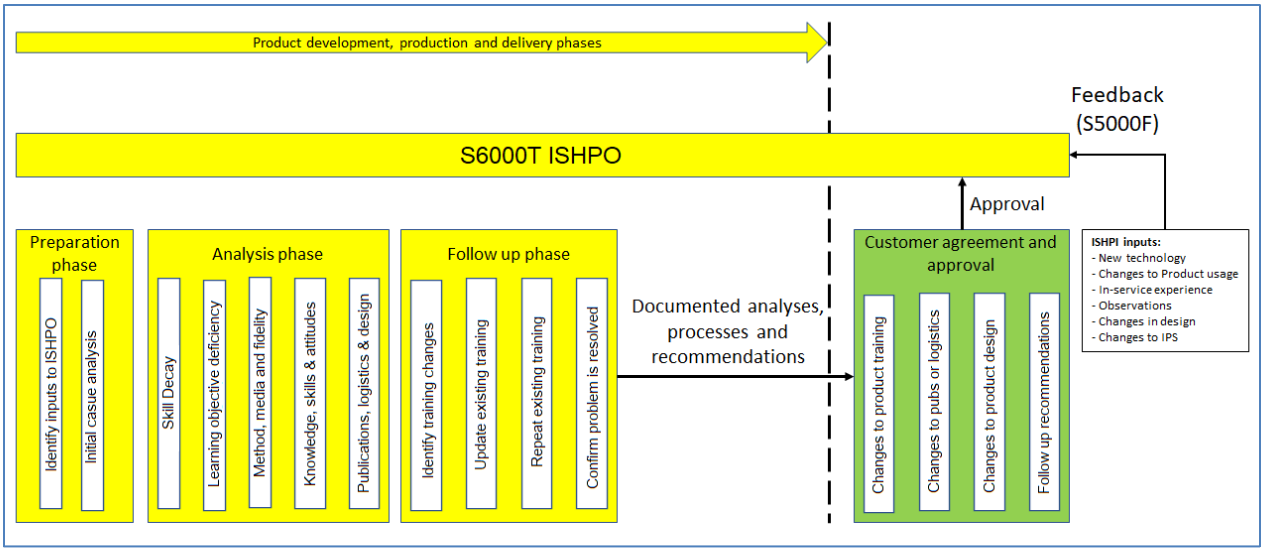
Each of these phases follows a detailed logical process. As the analyst works through the phases, it is recommended that:

- the customer and supplier agree a schedule for the ISHPO activities
- all ISHPO activities and outcomes be clearly documented
- the ISHPO activities and outcomes be discussed by the customer and supplier

Once the customer and supplier agree on the outcomes of the ISHPO process, approval to proceed can be given and the recommendations from the ISHPO follow-up phase implemented. Refer to Fig 1.

Note

Feedback can originate at any point in the ISHPO process. Refer to S5000F.



ICN-S6000T-B6865-00080-002-01

Fig 1 S6000T ISHPO process

2.1 Measuring human performance

To determine whether the ISHPO process has been effective, it is essential that there is a means to measure the human performance both before and after the process. For example, the Key Performance Indicator (KPI) can provide the required measurements.

2.2 Inputs

Fig 1 shows that, when a Product is in-service, there are numerous through-life opportunities to gather information and feedback for the ISHPO process. During ISHPO, this information and feedback is used to determine whether the current Training Needs Analysis (TNA), together with the training designed from that TNA, are still valid. If they are no longer valid, the TNA and training need to be reanalyzed.

As the amount of information and feedback increases, the probability of achieving high quality ISHPO results also increases. Therefore, all possible in-service data and customer feedback, from Product usage, maintenance, and training, are used as inputs to ISHPO.

Note

The availability of the inputs listed in Para 2, and shown in Fig 1, are not exhaustive or decisive prerequisites. The ISHPO logic process can also be used with small or missing in-service data inputs.

The ISHPO process can also include other human performance related analysis data, which comes from similar Products with a variety of histories and backgrounds.

2.3 Aim of the ISHPO process

The primary aim of the ISHPO process is to improve the operators' and maintainers' human performance when using the product to support a mission. ISHPO can also help to decrease the operational costs of the Product.

It is essential that the ISHPO process is performed more than once during the Product's lifecycle. After the start of an in-service phase, the setup of an initial ISHPO analysis consumes the major preparation and analytical effort. These initial investigations are the prerequisite to performing later analysis on the human performance aspects of the Product's lifecycle.

During the early life of the Product, the initial ISHPO decisions that will be made are likely to be conservative. Experience shows that better optimization can be identified by performing further ISHPO activities throughout the Product's lifecycle. The result of this will be that:

- human performance levels are increased
- in-service training costs are decreased

2.4 Operations

Many Products are operated by customers worldwide and these can be supported by a variety of training products throughout a Product's lifecycle. These training products can vary widely in content and detail but, in all cases, valid and up-to-date information and feedback is required for each implementation of ISHPO. Like the training, this information and feedback can vary widely from one customer to another.

Although training systems can vary widely, the creation of each Product training system is driven by several information sources and Analysis, Design, Development, Implementation and Evaluation (ADDIE) based processes, including but not limited to:

- Analysis:
  - training situation analysis. Refer to Chap 3.2.
  - mission, job, and task analysis. Refer to Chap 3.3.
  - task selection. Refer to Chap 3.4.
  - training task analysis. Refer to Chap 3.5.
  - training strategy. Refer to Chap 3.6.
  - business objects. Refer to Chap 3.7.
- Design:
  - target audience description. Refer to Chap 4.2.
  - learning gaps. Refer to Chap 4.3.
  - learning Objectives (LO). Refer to Chap 4.4.
  - assessment strategy. Refer to Chap 4.5.
  - instructional strategy . Refer to Chap 4.6.
  - media selection. Refer to Chap 4.7.
  - LO sequencing. Refer to Chap 4.8.

- training system alternative . Refer to Chap 4.9.
- training system requirements. Refer to Chap 4.10.
- curriculum outline approval and release. Refer to Chap 4.11.
- business objects. Refer to Chap 4.12.

It is recommended that, during the ISHPO process:

- all inputs be analyzed against the Product design data, the original TNA and the training design information
- all ISHPO activities are clearly documented for use during the current and any subsequent ISHPO events

3 ISHPO logic

The ISHPO process is described by a logical flow diagram, which guides the analyst through a series of decisions and actions. By following the flow diagram, the ISHPO inputs are defined, any potential human performance improvements are identified, and appropriate action is taken to make these improvements. The Complete ISHPO Flow diagram is shown in Fig 2 and Fig 3.

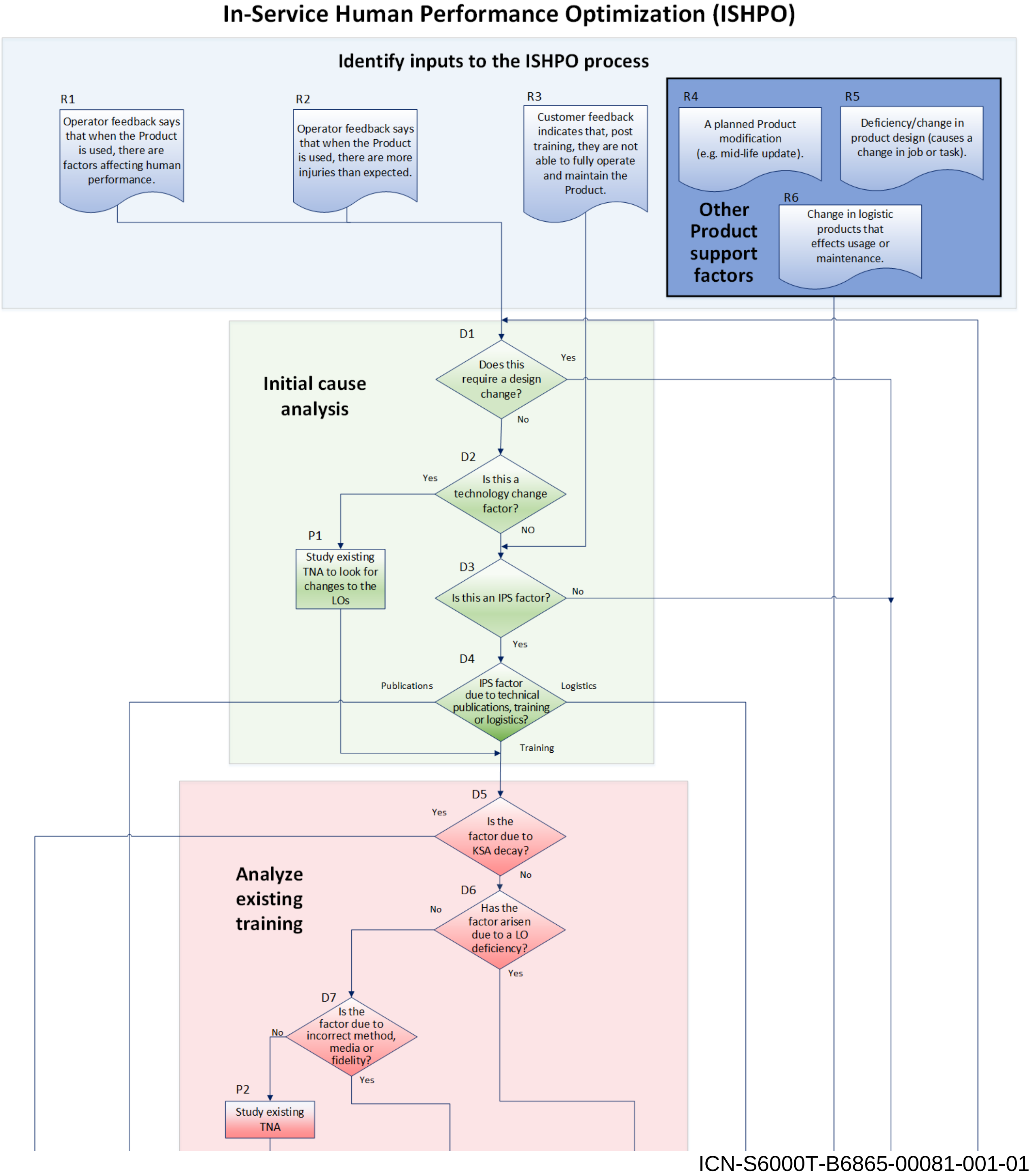
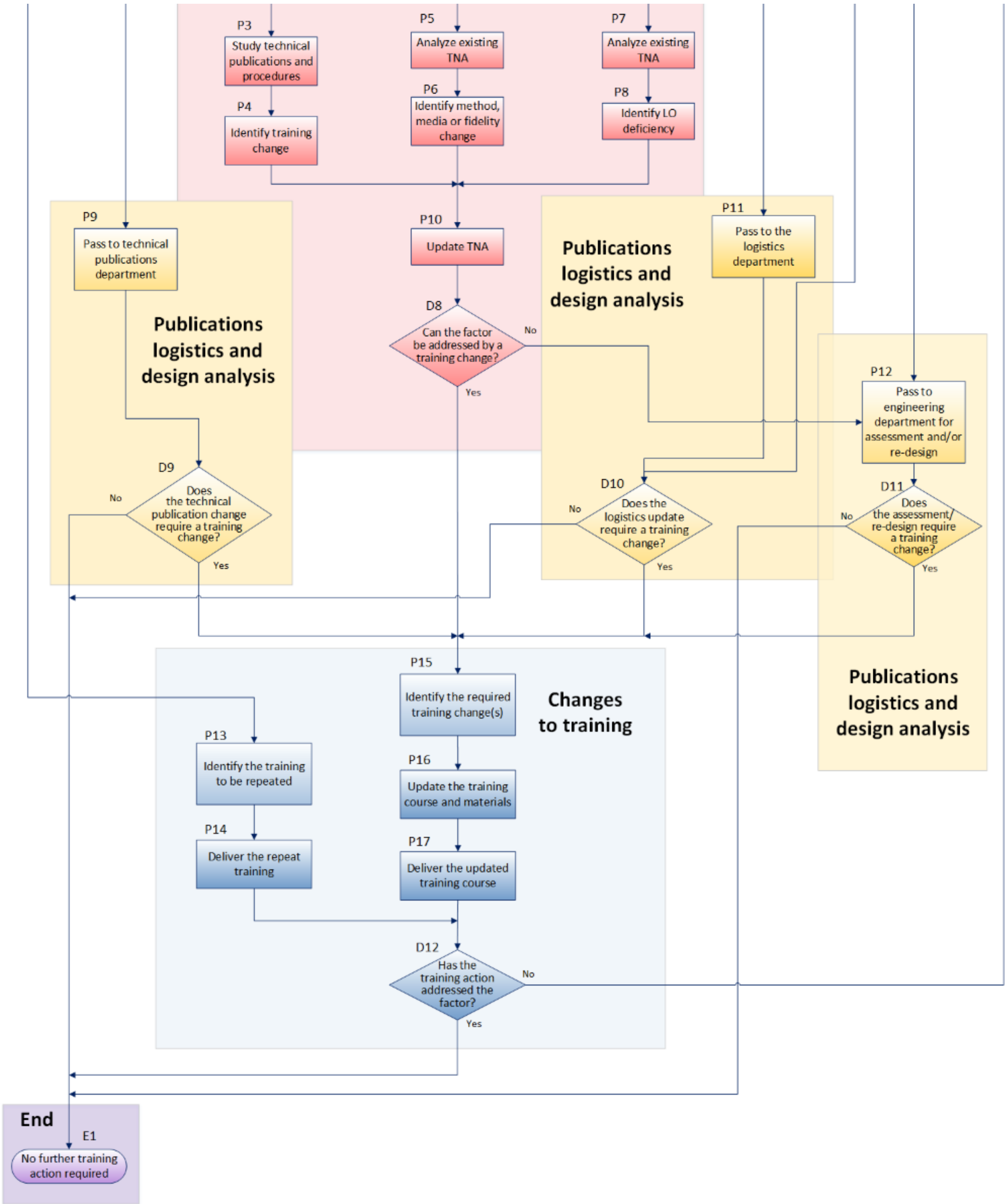


Fig 2 S6000T ISHPO Flow Diagram (Sheet 1 of 2)



ICN-S6000T-B6865-00082-001-01

Fig 3 S6000T ISHPO Flow Diagram (Sheet 2 of 2)

Chapter 5.2

In-Service Human Performance Optimization - Preparation phase

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 5.1	In-Service Human Performance Optimization - Introduction
Chap 5.3	In-Service Human Performance Optimization - Analysis phase
S3000L	International procedure specification for Logistic Support Analysis (LSA)
S4000P	International specification for developing and continuously improving preventive maintenance
S5000F	International specification for in-service data feedback

1 Introduction

The In-Service Human Performance Optimization (ISHPO) preparation phase is a detailed examination of the external inputs to the ISHPO process. The primary aim of this phase is to identify contributing factors affecting human performance that can be resolved by a change to the Product training.

The complete ISHPO process follows a comprehensive and logical flow diagram (refer to Chap 5.1) that guides the analyst through a series of decisions and actions. By following the flow diagram, the contributing factor is defined, the cause is identified and the appropriate action to rectify it is completed. The ISHPO preparation phase is designed to examine the external inputs to the ISHPO process and confirms that the contributing factor requires a change to any of the following:

- Product design
- logistics and Product support
- technical publications
- Product training

The ISHPO preparation phase is included in the ISHPO overarching flow diagram, in which the sections highlighted as "Identify inputs to the ISHPO process" and "Initial cause analysis" cover the ISHPO preparation phase. Refer to Fig 1.

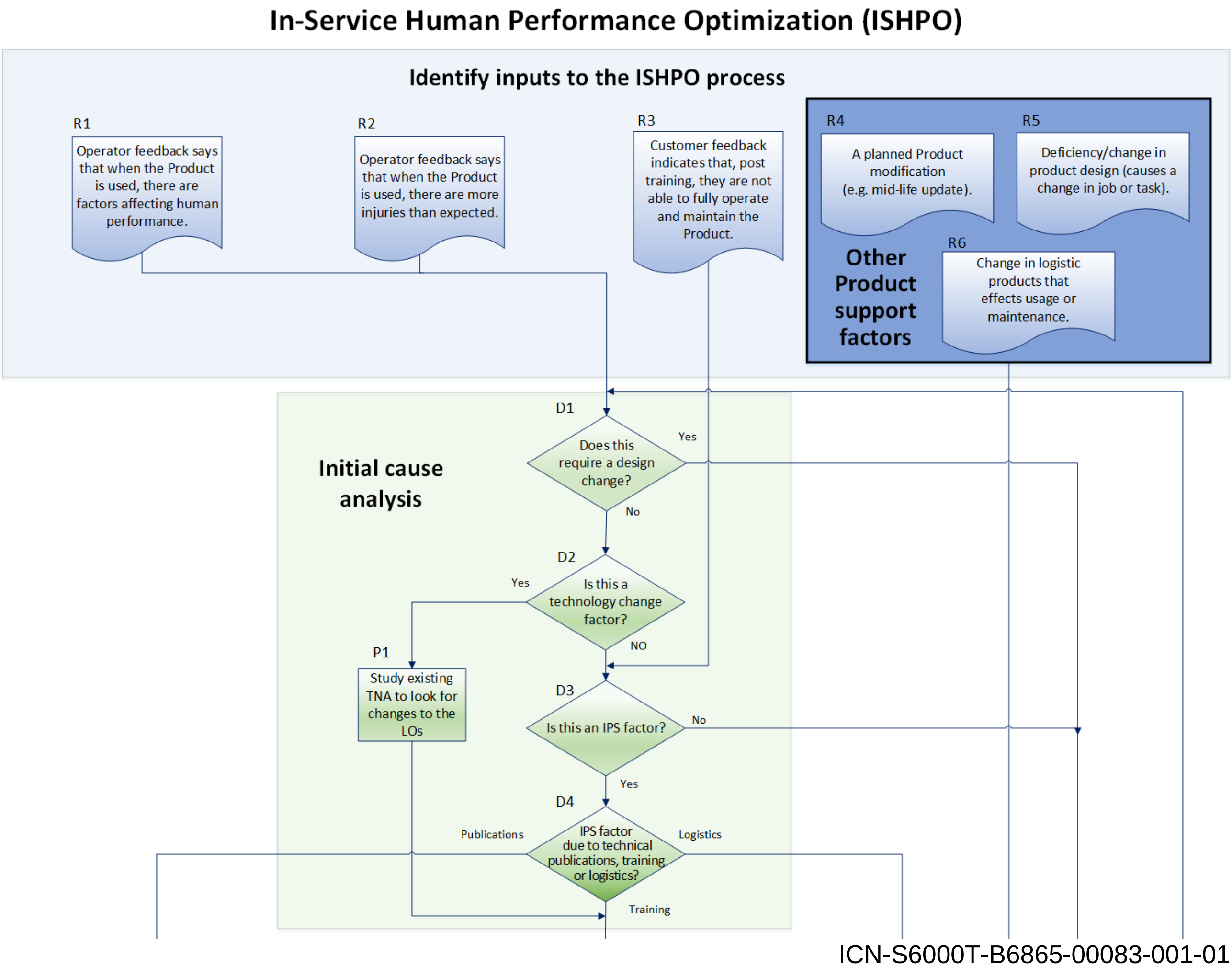


Fig 1 ISHPO Preparation Phase

2 Input/trigger

The ISHPO preparation phase will be required each time that the ISHPO process is carried out. The ISHPO process can be required for several reasons including, but not limited to:

- operator feedback
- maintenance feedback
- system upgrade
- change in Product design
- change in Product use

3 The ISHPO preparation phase

The ISHPO preparation phase guides the analyst through a series of decisions and actions. These decisions and actions will identify the primary Product support area that needs to be addressed (refer to Para 4 and Para 5).

3.1 Output

The output of the ISHPO preparation phase will be a requirement for either:

- a Product design change (refer to Para 5.2)
- a Product support change (refer to Para 5.5)

- a publication update (refer to Para 5.5)
- a training update or change (refer to Para 5.4 or Para 5.5)

4 Identify inputs to the ISHPO Process

4.1 Introduction

The Integrated Product Support (IPS) department will routinely gather data and feedback to ensure that the support for a Product is meeting the Product and customer needs. At the beginning of an ISHPO preparation phase, all data that has been gathered and fed back to the IPS department is examined to identify any input that affects human performance.

4.1.1 Inputs related to human performance

All data and feedback inputs, which are related to human performance, are examined to determine whether they are candidates for further analysis during the ISHPO process. Those aspects that do not require further analysis are disregarded and are not used as inputs to the ISHPO process.

Any data and feedback inputs related to human performance, which do require further consideration, become the inputs to the ISHPO preparation phase.

The ISHPO preparation phase also requires the collection of other information and data sources that will be used during the ISHPO process. This will include, but is not limited to:

- existing Training Needs Analysis (TNA)
- current training course syllabus and lesson plans
- current course materials
- current Product publications
- current Product support procedures

Where parts of the existing analytical training design and development process do not exist or cannot be verified, subsequent analysis activities must be examined for areas of potential compensation. This can be supported by utilizing operating experience with the Product.

4.1.2 Process

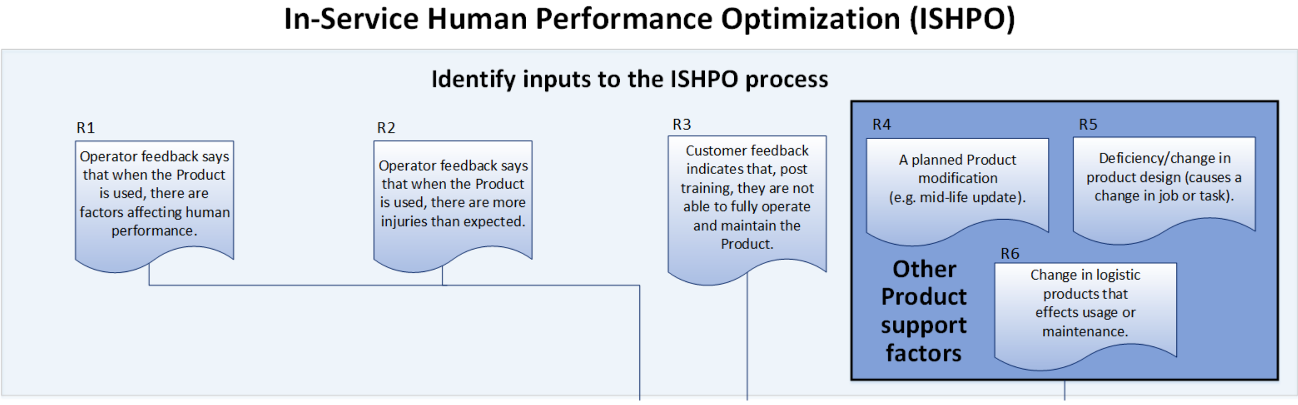
The identify inputs to the ISHPO preparation phase (refer to Fig 2) will guide the analyst through the following steps:

1. Identify the inputs raised by operators, maintainers, or customer feedback (refer to Para 4.2 and Para 4.3).

2. Identify the inputs that are due to planned Product design changes, system deficiencies or planned Product support changes (refer to Para 4.4).

4.1.3 Output

The output of the identify inputs to the ISHPO preparation phase feeds the identified input, and its associated contributing factor, to the appropriate process in the initial cause analysis (refer to Para 5) or the ISHPO analysis phase (refer to Chap 5.3).



ICN-S6000T-B6865-00084-001-01

Fig 2 Identify Inputs to the ISHPO Preparation Phase

4.2 Operator feedback (R1 & R2)

An important part of the preparation phase includes identifying and confirming that there is a genuine contributing factor with the Product or system, or with the users and maintainers employed on the Product or system.

Operator feedback received from the users and maintainers could indicate that:

- when the Product is used, there are frequent factors that affect the Product's operational efficiency (R1), which could also lead to an increase in maintenance
- when the Product is used and maintained, there are more than the expected number of injuries (R2)

All the above conditions will require investigation, as there may be a need to change the Product training.

4.2.1 Output

The outputs of R1 and R2 feed into decision box D1 of the initial cause analysis to identify if a Product design change is required (refer to Para 5).

4.3 Customer feedback regarding training (R3)

Feedback from the customer about the Product or system may indicate that the operators and maintainers are unable to operate and maintain the Product, even though they have been fully trained (R3). This is likely to be caused by an IPS deficiency.

Feedback that identifies a possible deficiency in the IPS should be investigated as soon as possible to determine if the IPS contributing factor requires a change or update to the Product training.

These contributing factors should be directed to the IPS team (D3), who will examine the factor and take the necessary action to either update the technical publications, training, or logistics, or pass it to engineering, who will look at options to redesign the Product.

4.3.1 Output

The output of R3 is a feed into decision box D3 of the initial cause analysis to determine if the contributing factor is caused by the IPS (refer to Para 5).

4.4 Product/system changes (R4, R5, R6)

A Product may be subject to modifications, Product design changes, procedural changes, and Product support changes throughout its lifecycle. These changes often occur at planned times in response to customer requests, Product improvements and midlife upgrades.

Other S-Series IPS specification optimizations can also be an input to the ISHPO process (refer to Chap 5.1) and each can trigger a requirement to change or improve the human performance when operating or maintaining the Product.

Product modifications, Product updates and changes to Product support, can require a change to the Product training. In the identify inputs section of the flowchart (refer to Fig 2), R4, R5 and R6 center on examining if there has been a change to the Product design or Product support and how the Product or system is currently being maintained and used. For example, changes could be caused by the requirement for additional maintenance, the system being used in a new environment, or new added functions. These changes could have come from S3000L, S4000P or S5000F specifications regarding maintenance and availability optimization.

4.4.1 Output

The outputs of R4, R5 and R6 feed into decision D10 of the ISHPO analysis phase to identify if a training change is required (refer to Chap 5.3).

5 Initial cause analysis

5.1 Introduction

The initial cause analysis is the first phase after a genuine problem with the Product has been identified.

The initial cause analysis uses the ISHPO flow diagram’s decision logic to examine feedback from operators, maintainers, and customers. This analysis will lead to an engineering change, a logistic change, changes to the technical documentation, or additional and/or revised training.

A thorough understanding of the Product, and how it is used, is essential to be able to perform the analysis effectively. While working through Fig 1, it is important to follow the steps in the correct order shown in the flow diagram. This will ensure that the result is as accurate as possible.

5.1.1 Input/trigger

The input and trigger to the initial cause analysis is an output from the identified inputs to the ISHPO preparation phase, confirming that there is a confirmed contributing factor with the system and/or Product.

5.1.2 Process

The initial cause analysis (refer to Fig 3) will guide the analyst through the following steps:

1. Identifying that the contributing factor requires a Product design change (refer to Para 5.2).
2. Identifying that the contributing factor is caused by a technology change.
3. Studying the existing TNA to identify changes to the Learning Objectives (LO) (refer to Para 5.4).
4. Identifying that the source of the contributing factor is an IPS issue. Refer to Para 5.5.
5. Determine if the IPS issue is due to technical publications, logistics or training (refer to Para 5.5).

5.1.3 Output

The output of the initial cause analysis is a decision that has identified if the contributing factor is caused by technical publications, logistics or training.

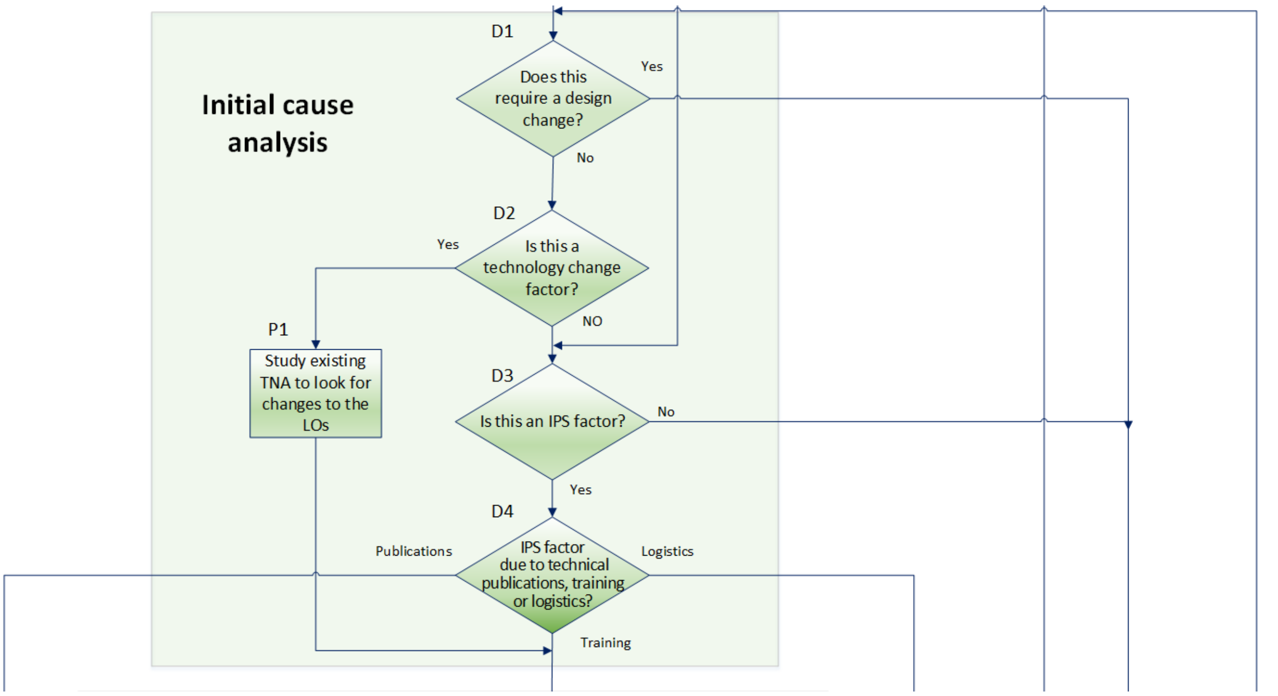


Fig 3 Initial cause analysis

5.2 Product design change (D1)

Sometimes a Product's design and the user requirements will change during a Product's development. Also, the user could be operating or maintaining the Product in a different way than it was originally designed or intended. These circumstances may require a change in the Product's design.

Decision D1 asks whether the contributing factor requires the Product engineering team to make a change to the Product design.

5.2.1 Output

If the contributing factor requires a change to the Product design, the contributing factor must be addressed by the Product engineering team (refer to Chap 5.3).

If the contributing factor does not require a change to the Product design, proceed to decision D2. Refer to Para 5.3.

5.3 Technology change (D2)

During the design phase of a Product, it is possible that the technologies used in the Product can change. If the person operating or maintaining the Product is unfamiliar with this new technology, it can have an impact on the required training . Also, if the original end-user changes, the new end-user could be unfamiliar with the technologies used in the Product.

Decision D2 asks if there is a technology change issue, due to either a change of technology in the Product or with the end-user. This change can occur during initial development or during a Product update and can cause problems for the user.

5.3.1 Output

If the contributing factor is due to a technology issue, proceed to process P1 (refer to Para 5.4), and study the existing TNA to ensure it meets the requirements of the current end-user.

If the problem is not due to a technology issue, proceed to decision D3 to confirm that the source of the factor is an IPS issue. Refer to Para 5.5.

5.4 Change of learning objectives (P1)

If the original TNA was performed early in the development process of a new Product, it is possible to miss some technology requirements or problems that only become evident later in the Product's lifecycle. The cause of this can be due to several reasons including, but not limited to:

- unclear requirements early in the system development
- the system being used in new ways that were not originally anticipated
- changes in the technology used to meet the Product's requirements

The analyst must study the existing TNA to identify any changes to the LOs that are required to address the technology change contributing factor (P1).

5.4.1 Output

If the analysis of the TNA indicates that changes are required to the LOs, this must be addressed during the ISHPO analysis phase (D6). Proceed to Chap 5.3 to define the training issue.

5.5 IPS issue (D3 & D4)

It is possible that the contributing factor identified in the feedback is caused by a deficiency in the Product's IPS. The deficiency could be in the:

- Product technical publications (ie, user manuals, maintenance manuals, etc)
- Product support (ie, spares, repair capability, supply chains, etc)
- Product training (ie, operator training, maintenance training, etc)

The analyst must study the problem to confirm that the source of the contributing factor is an IPS issue (D3).

5.5.1 Output

If the source of the contributing factor is an IPS issue, the analyst must investigate the factor further to determine whether it is caused by the Product technical publications, training or logistics (D4). After this analysis, the analyst should proceed, as follows, to the relevant section in Chap 5.3:

- if the issue is due to technical publications (eg, incorrect procedures etc), it should be passed to the publications team (P9)
- if the issue is due to training (eg, skill decay, training method etc), it needs to be further analyzed in the define training issue phase (D5)
- if logistics is the cause (eg, special tools that do not fit properly, lack of spares, etc) it should be passed to the logistics team (P11)

If the output of decision D3 indicates that the issue is not due to an IPS issue, the Product design needs to be examined to see if a design change can address the problem. The problem should now be passed to the Product engineering team (P12).

Chapter 5.3

In-Service Human Performance Optimization - Analysis phase

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 5.1	In-Service Human Performance Optimization - Introduction
Chap 5.2	In-Service Human Performance Optimization - Preparation phase
Chap 5.4	In-Service Human Performance Optimization - Follow-Up phase

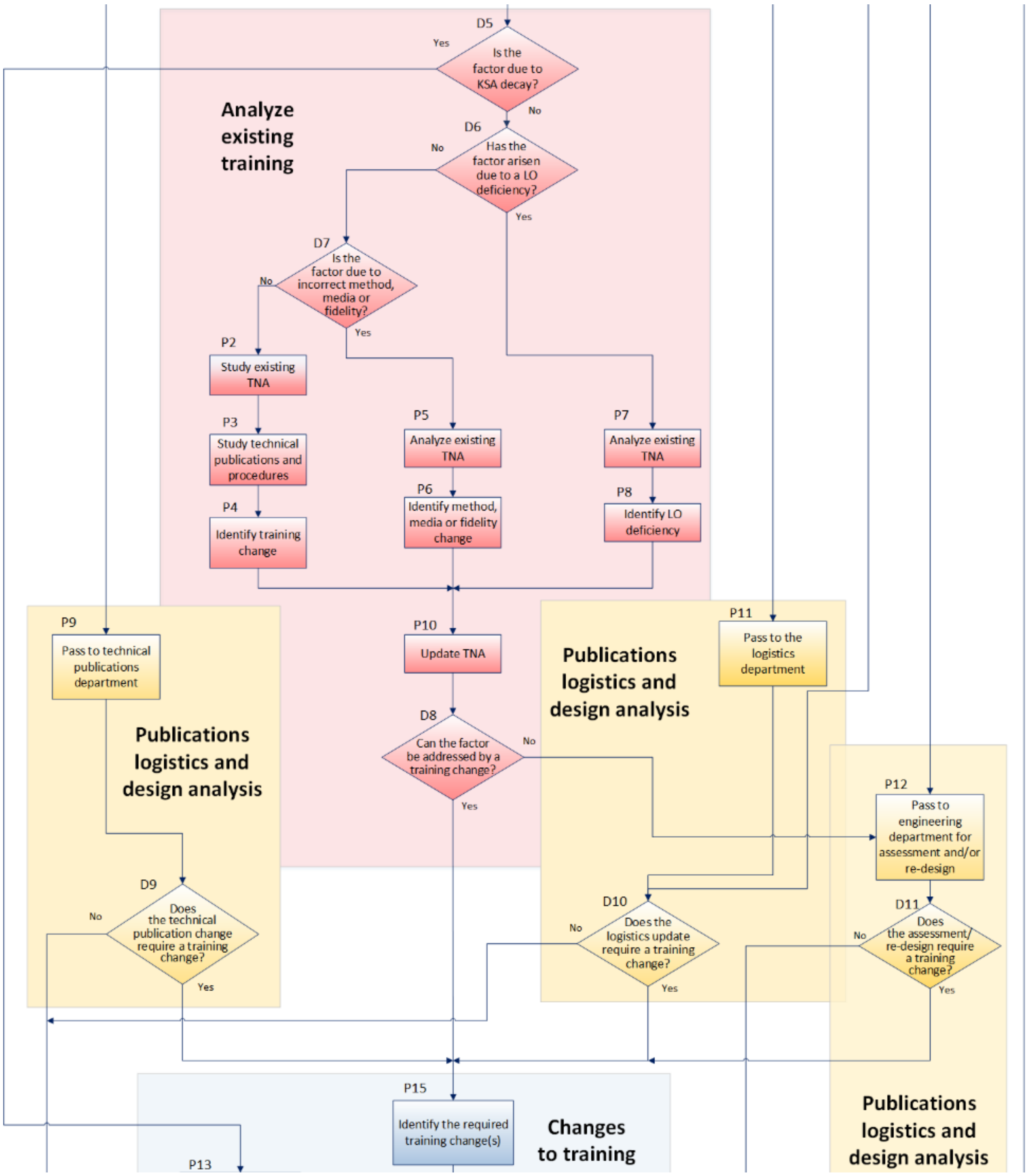
1 Introduction

The In-Service Human Performance Optimization (ISHPO) analysis phase is a detailed analysis of the Product training and its associated technical publications and logistics. The analysis looks at how the current Product training, technical publications and logistics can be contributing factors to an Integrated Product Support (IPS) issue identified and discussed in the ISHPO preparation phase (refer to Chap 5.2).

The complete ISHPO process follows a comprehensive and logical flow diagram that guides the analyst through a series of decisions and actions. By following the flow diagram, the IPS issue is defined, the cause of the issue is identified and the appropriate action to rectify the issue is completed. The complete ISHPO flow diagram is shown in Chap 5.1.

The ISHPO analysis phase is designed to identify the cause of any contributing factors and to update the Training Needs Analysis (TNA), technical publications, logistics or design accordingly.

The ISHPO analysis phase is displayed in the ISHPO flow diagram (refer to Chap 5.1). The sections highlighted as Analyze Existing Training and Publications, Logistics and Design Analysis cover the ISHPO analysis phase, as shown in Fig 1.



ICN-S6000T-B6865-00086-001-01

Fig 1 ISHPO analysis phase

2 Input/trigger

The ISHPO analysis phase will only be required if the ISHPO preparation phase (refer to Chap 5.2) has identified that the contributing factor is caused by training, technical publications, logistics or Product design.

3 The ISHPO analysis phase

The ISHPO analysis phase guides the analyst through a series of decisions with resultant actions. These decisions and actions will identify the cause of the contributing factor and prescribe appropriate rectification (refer to Para 4 and Para 5).

3.1 Output

The output of the ISHPO analysis phase will be either:

- additional training (refer to Chap 5.4)
- a change to the TNA and training, plus a decision as to whether the training change will fix the contributing factor (refer to Para 4)
- a technical publication change, possibly with an associated training update (refer to Para 5)
- a logistics change, possibly with an associated training update (refer to Para 5)
- a Product design change, with an associated training update (refer to Para 5)

4 Analyze existing training

4.1 Analysis objectives

To analyze existing training, S6000T uses a detailed process, specifically designed to identify any contributing factors with the training and any subsequent improvements. A thorough understanding of the Product, and how it is used, is essential to be able to perform this analysis effectively. While working through Fig 1, it is important to follow the steps in the correct order shown in the flow diagram. This will ensure that the result is as accurate as possible.

Prior to the analyze the existing training process, the ISHPO preparation phase will have identified whether the cause of the contributing factor is due to Product training, technical publications, logistics or Product design (refer to Chap 5.2). If the cause is due to Product training, analysis of the existing training will examine the current TNA and training product to identify where changes are required.

Analyzing the existing training follows the flow diagram's decision logic to analyze the following:

- Knowledge, Skill and/or Attitude (KSA) decay
- Learning Objective (LO) deficiencies in the TNA
- training methods, media and fidelity used for training
- technical publications and procedures used during training

Having identified any required changes, the analysis determines whether a change to the training product will address the contributing factor. If the answer is yes, the training will be updated (refer to Chap 5.4).

4.1.1 Input/trigger

The input and trigger to the analyzing the existing training process is an output from the ISHPO preparation phase that identifies a contributing factor with the Product training.

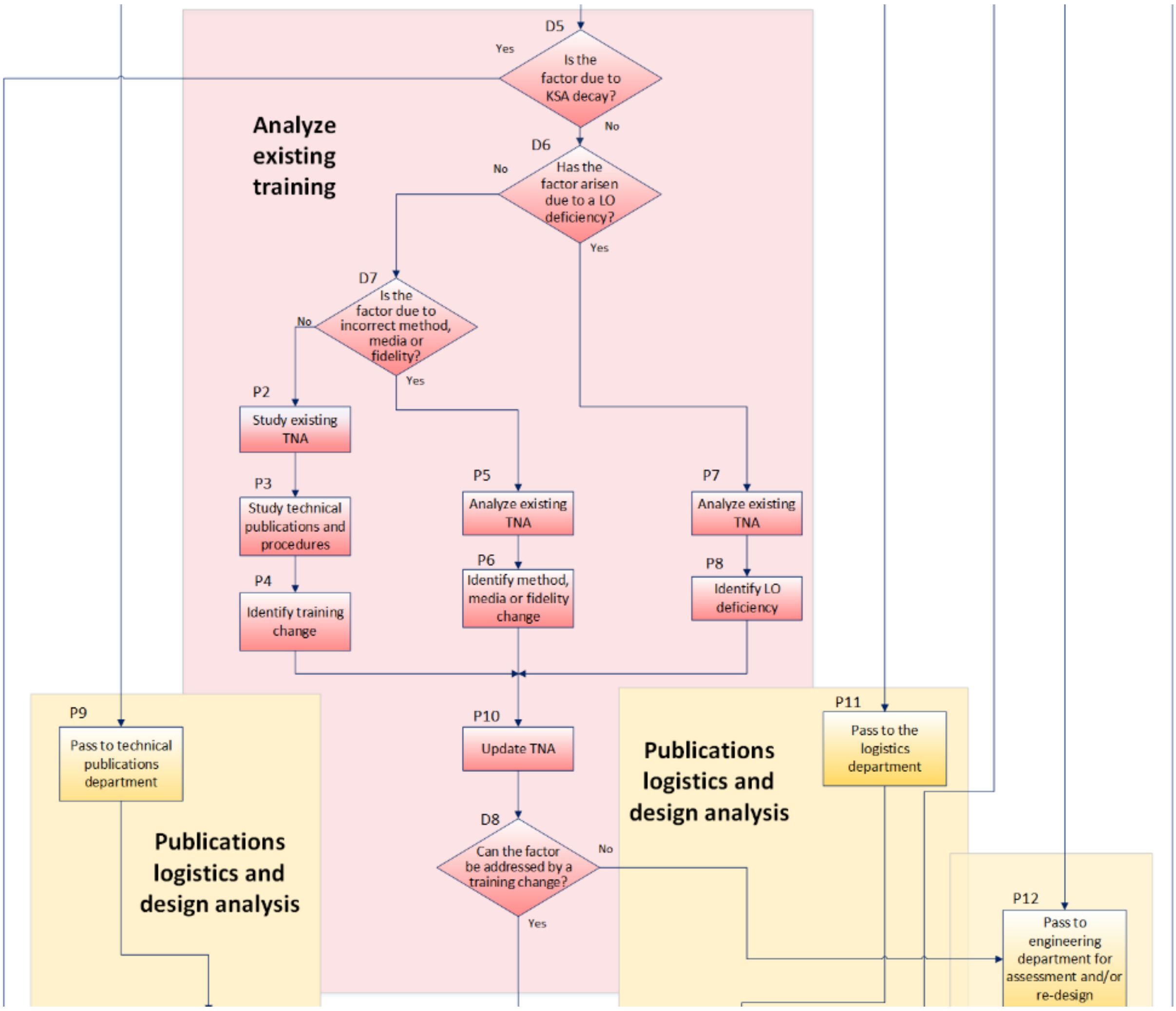
4.1.2 Process

The analysis will guide the analyst through the following steps:

1. Identify whether the contributing factor is caused by KSA decay (Para 4.2).
2. Identify whether the contributing factor is caused by a LO deficiency in the TNA (Para 4.3).
3. Identify whether the contributing factor is caused by not using the most appropriate method, media and/or fidelity (Para 4.4).
4. Identify whether the contributing factor is caused by the procedures to be learnt or the technical publications used to do those procedures (Para 4.5).
5. Update the TNA and confirm the updates address the contributing factor (Para 4.6).

4.1.3 Output

The output of the analysis is an updated TNA and a decision as to whether a training change will address the contributing factor.



ICN-S6000T-B6865-00087-001-01

Fig 2 Training problem analysis

4.2 Knowledge, skill, and attitude decay (D5)

The first decision to be made (D5) is whether the identified contributing factor is due to KSA decay.

KSA decay can be a serious problem when the KSA acquired during training are not used for a prolonged period. Ideally, training should be delivered as close as possible to the time when the KSA will be required to do the task(s). However, the following are potential reasons why training sometimes do not occur close to the application of KSA:

- the trainees are only available on certain dates
- there are delays to the Product in-service date
- after training, the trainees are re-tasked or relocated for a prolonged period
- a particular task is done infrequently

KSA decay can have a profound effect on a person's ability to do the tasks they were trained for. The degree to which their ability is affected will depend upon:

- the task difficulty
- the task frequency
- the length of time not using the KSA
- tasks being added after initial training was performed

4.2.1 Output

If the analysis indicates that KSA decay is the cause of the contributing factor, then repeat training will probably be required. The process to identify training that must be repeated is covered in Chap 5.4.

If the analysis indicates that KSA decay is not the cause of the training contributing factor, proceed to Para 4.3.

4.3 Learning objective deficiency (D6)

If KSA decay is not the cause of the contributing factor, the next step (D6) determines whether the identified factor is due to a deficiency in the LO.

When the original TNA was carried out, it is possible that the task list was incomplete or has changed since the TNA was completed. This can be due to a number of reasons including, but not limited to:

- changes in Product design
- delays in Product development
- changes to procedures

The LO must be comprehensive to ensure the training is appropriate, covers all tasks required and meets the Product's operational objectives. Any deficiencies in the LO will result in trainees having a KSA gap and being insufficiently trained to do the tasks.

4.3.1 Output

If the analysis indicates that a deficiency in the LO is the cause of the training contributing factor, the current task inventory and the TNA will need to be analyzed (P7) and the LO deficiency identified (P8). After this analysis is complete, proceed to Para 4.6.

If the analysis indicates that there is not a deficiency in the LO, proceed to Para 4.4.

4.4 Inappropriate method, media, or fidelity (D7)

During the TNA, it is important to select the most appropriate methods and media to achieve the learning outcomes of a training event. If the training does not use the correct methods and/or media, then the trainees will find it more difficult to achieve the LO.

During the initial TNA, the analyst will select the most appropriate method and media available at the time. However, the most appropriate method and media can change due to several reasons including, but not limited to:

- changes to the Product design
- changes to tasks and procedures
- changes to the available budget

The media used in the training must also meet the fidelity requirements. If the media does not correctly replicate the tasks, or is over-complicated, the trainees can become confused about what they need to do and learn.

4.4.1 Output

If the analysis indicates the current method and/or media are no longer the most appropriate, or the fidelity of the media is incorrect, then the TNA will need to be analyzed (P5) and the new, method, media or fidelity identified (P6). After this analysis is complete, proceed to Para 4.6.

If the analysis indicates that the current method, media, and fidelity are appropriate and not the cause of the training contributing factor, proceed to Para 4.5.

4.5 Technical publications and procedural issues (P2)

During training it is likely that several procedures will need to be practiced by the trainees. If there are problems with the technical publications used for training, or the procedures selected for training are not trained correctly, it can result in trainees not acquiring the necessary KSA to do the tasks.

During the initial TNA, the analyst will select the procedures to be trained, based on the Difficulty, Importance and Frequency (DIF) analysis. The instructors will then interpret and use the procedures to train the trainees. However, the most appropriate procedures to be trained can change, or there can be problems with the procedural steps in the technical publications, due to several reasons including, but not limited to:

- changes to the Product design
- changes to a task's DIF
- procedural problems being identified after the publication was issued

4.5.1 Outputs

If the analysis indicates that the technical publications or procedures are the cause of the contributing factor, then the TNA will need to be analyzed (P2) and the technical publications and procedures will also need to be studied (P3). The analyst must then identify what is causing the contributing factor (P4) and decide whether:

- the procedures being trained are the most appropriate
- the procedures being trained are being interpreted and/or done incorrectly

After this analysis is complete, proceed to Para 4.6.

4.6 Update training needs analysis (P10)

While analyzing the existing training , the analyst will have followed the flowchart and identified the contributing factor that is contributing to the overall IPS issue.

The next step is to update the TNA, and possibly the training design documents, to address the training factor (P10). This update could involve a change to:

- the LO and their associated KSA
- the method, media and fidelity chosen to train the LO
- the technical publications and selected procedures used during training

Once the TNA has been updated, this new TNA is checked to ensure that the contributing factor can be fully addressed by only changing the training (D8).

4.6.1 Outputs

If the check of the TNA indicates the training factor can be fully addressed by only changing the training, proceed to Chap 5.4 and update the training course and materials.

If the check of the TNA indicates the training factor cannot be fully addressed by only changing the training, they will need to pass the issue to the Product design/engineering team for assessment and possible redesign of the Product (P12) (refer to Para 5.4).

5 Publications, logistics and design analysis

5.1 Introduction

The technical publications, logistics and design analysis uses the ISHPO flow diagram's decision logic to deal with issues that require:

- a technical publication change (refer to Para 5.2)

- a logistics change (refer to Para 5.3)
- a Product design change (refer to Para 5.4)

A thorough understanding of the system and/or Product, and how it is used, is needed to perform this analysis correctly. It is important to follow the steps in the correct order. This will ensure that the result is as accurate as possible.

Prior to the ISHPO analysis phase, the ISHPO preparation phase will have identified if the cause of the contributing factor is due to training, technical publications, logistics or Product design (Refer to Chap 5.2). If the cause is due to technical publications, logistics or Product design, the ISHPO preparation phase passes the issue to the appropriate department or team. The ISHPO analysis phase then addresses the technical publications, logistics or Product design issue and, subsequently, identifies any training updates that will be required (refer to Chap 5.4).

5.1.1 Input

The input and trigger to the publications, support products and design analysis is an output from the ISHPO preparation phase that identifies there is a contributing factor with the publications, logistics or Product design (refer to Chap 5.2).

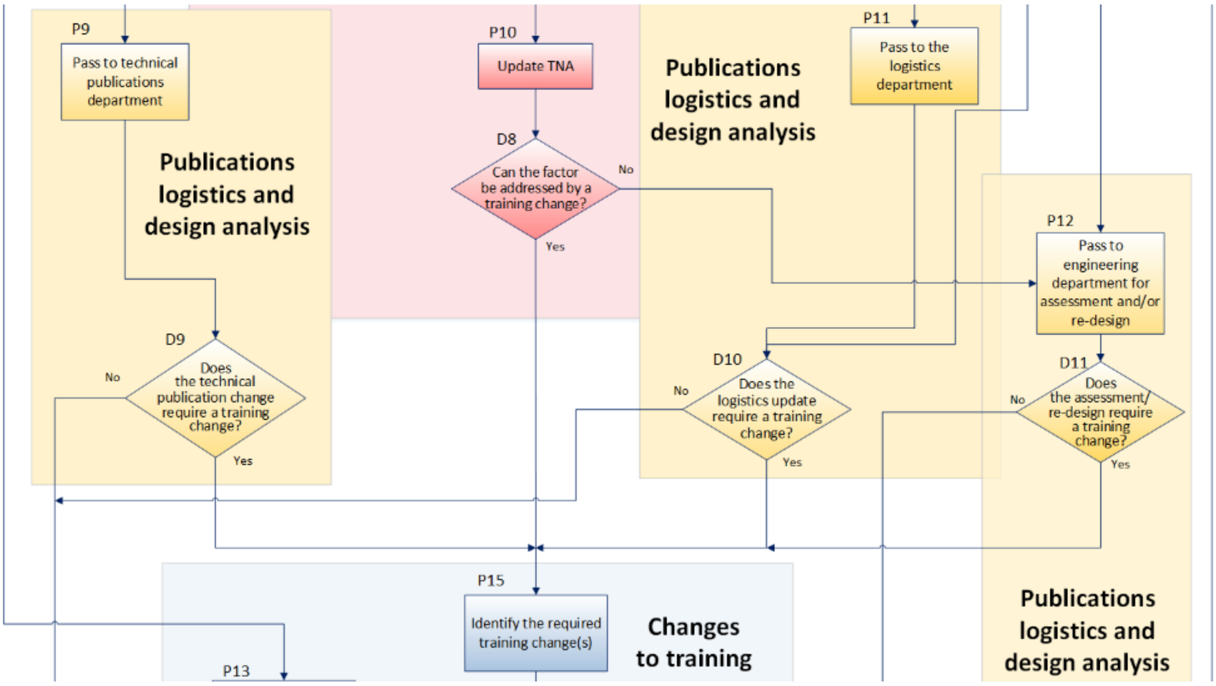
5.1.2 Process

The publications, logistics and design analysis will guide the analyst through the following steps:

1. Pass the problem to the publications, logistics or Product design department.
2. Await the publications, logistics or Product design potential change.
3. Identify whether the publications, logistics or Product design change requires a change to the training.

5.1.3 Output

The output of the publications, logistics and design analysis is a decision as to whether a training change is required or not.



ICN-S6000T-B6865-00088-001-01

Fig 3 Publications, logistics and design analysis

5.2 Technical publications (P9)

Technical publications are vital for the correct, efficient use and training which supports a Product. Any anomalies or ambiguities can result in incorrect training and subsequent, incorrect use and maintenance of the Product.

The ISHPO preparation phase will identify if the contributing factor with the Product is caused by one or more technical publications.

If a contributing factor is caused by a technical publication issue, the technical publications department or team is notified about the issue (P9). Once the factor in the technical publication has been corrected, the analyst will study the change and decide whether the change to the technical publication requires a change to the training (D9).

5.2.1 Output

If the change to the technical publication requires a change to the training, proceed to Chap 5.4 and update the training course and materials.

If the change to the technical publication does not require a change to the training, then no further action is required with the training (E1) and subsequent training events will follow the updated information/procedures in the technical publication.

5.3 Support products (P11)

Logistics and support are vital for the successful use of a Product, particularly in the long term, as well as for the training supporting that Product. Any logistic or support issues can put an extra burden or requirement on the user and maintainer, particularly if they have not been trained to deal with the issue.

The ISHPO preparation phase will identify if the contributing factor with the Product is caused by logistics or support products.

If a contributing factor with the logistics or support products has been identified, the logistics or Product support department or team will be notified about the issue (P11). Once the issue with the logistics or support product has been corrected, the analyst will study the change and decide whether this change requires a change to the training (D10).

5.3.1 Output

If the change to the logistics or support product requires a change to the training, proceed to Chap 5.4 and update the training course and materials.

If the change to the logistics or support product does not require a change to the training, then no further action is required with the training (E1) and subsequent training events will follow the existing training syllabus.

5.4 Design changes (P12)

There are certain Product issues that cannot be addressed by only making a change to the Product training, technical publications, or logistics. Some of these will require a change to the Product design. These changes can be either hardware and/or software changes.

During the ISHPO preparation phase (refer to Chap 5.2), the first decision (D1) asks if a Product design change is required. Also, in the ISHPO analysis phase, there is a decision that asks if the contributing factor can be addressed by only changing the training (D8).

If the ISHPO preparation phase or the ISHPO analysis phase determines that a Product design change is required, the Product/engineering design department or team will be notified about the issue (P12). Once the design change requirement has been analyzed and any necessary change made, the analyst will study the change and decide whether this requires a change to the training (D11).

5.4.1 Output

If a change to the Product design requires a change to the training, proceed to Chap 5.4 and update the training course and materials.

If a change to the Product design does not require a change to the training, then no further action is required with the training (E1) and subsequent training events will follow the existing training syllabus.

End of data module

Chapter 5.4

In-Service Human Performance Optimization - Follow-up phase

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 5.1	In-service human performance optimization - Introduction
Chap 5.2	In-service human performance optimization - Preparation phase
Chap 5.3	In-service human performance optimization - Analysis phase

1 Introduction

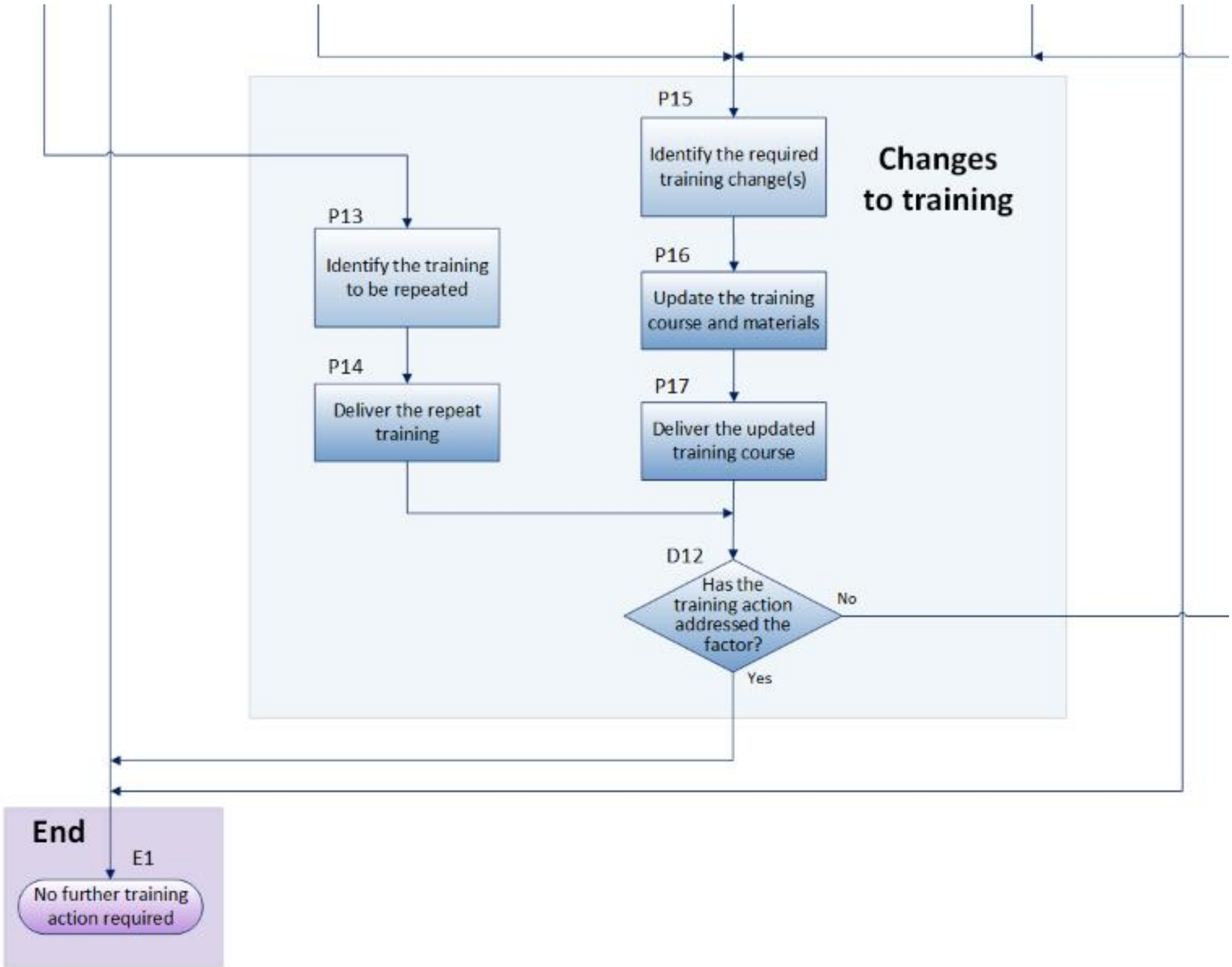
The In-Service Human Performance Optimization (ISHPO) follow-up phase is specifically designed to guide the training analyst and designer through the various training changes and activities that have been identified during the ISHPO analysis phase (refer to Chap 5.3).

The complete ISHPO process follows a comprehensive, logical, flow diagram that guides the analyst through a series of decisions and actions. By following the flow diagram, the IPS contributing factor is defined, the cause of the factor is identified, and appropriate action is taken to rectify the factor. The complete ISHPO flow diagram is shown in Chap 5.1.

The ISHPO follow-up phase follows the flow diagram's logic and address:

- identifying the training changes required and updating the training course and materials
- identifying any training that needs to be repeated
- delivering new or additional training
- checking that the new or additional training activities have addressed the contributing factor with the Product

The ISHPO follow-up phase is displayed in the ISHPO flow diagram (refer to Chap 5.1). The sections highlighted as Changes to training and End, cover the ISHPO follow-up phase (refer to Fig 1).



ICN-S6000T-B6865-00089-001-01

Fig 1 ISHPO Follow-up phase

2 Input/trigger

The ISHPO follow-up phase will only be required if the ISHPO analysis phase (refer to Chap 5.3) has identified that changes to the existing training or repetition of the existing training is required.

3 Follow-up phase

The ISHPO follow-up phase guides the analyst and designer through a series of actions and decisions. These actions and decisions are designed to address any issues, caused by the contributing factor, by updating or changing the Product training.

3.1 Output

The output of the ISHPO follow-up phase will be either:

- no further training action is required
- the training changes or actions have not fully addressed the Product issue and the contributing factor will need to be re-examined

4 Changes to training

4.1 Introduction

The follow-up phase, changes to training, is designed to specifically identify and make the training changes or activities required to address the contributing factor causing a Product issue. A thorough understanding of the system and/or Product, and how it is used, is needed to perform these activities correctly. It is important to go through the steps in the correct order shown in the flow diagram. This will ensure that the result is as accurate as possible.

4.1.1 Input/trigger

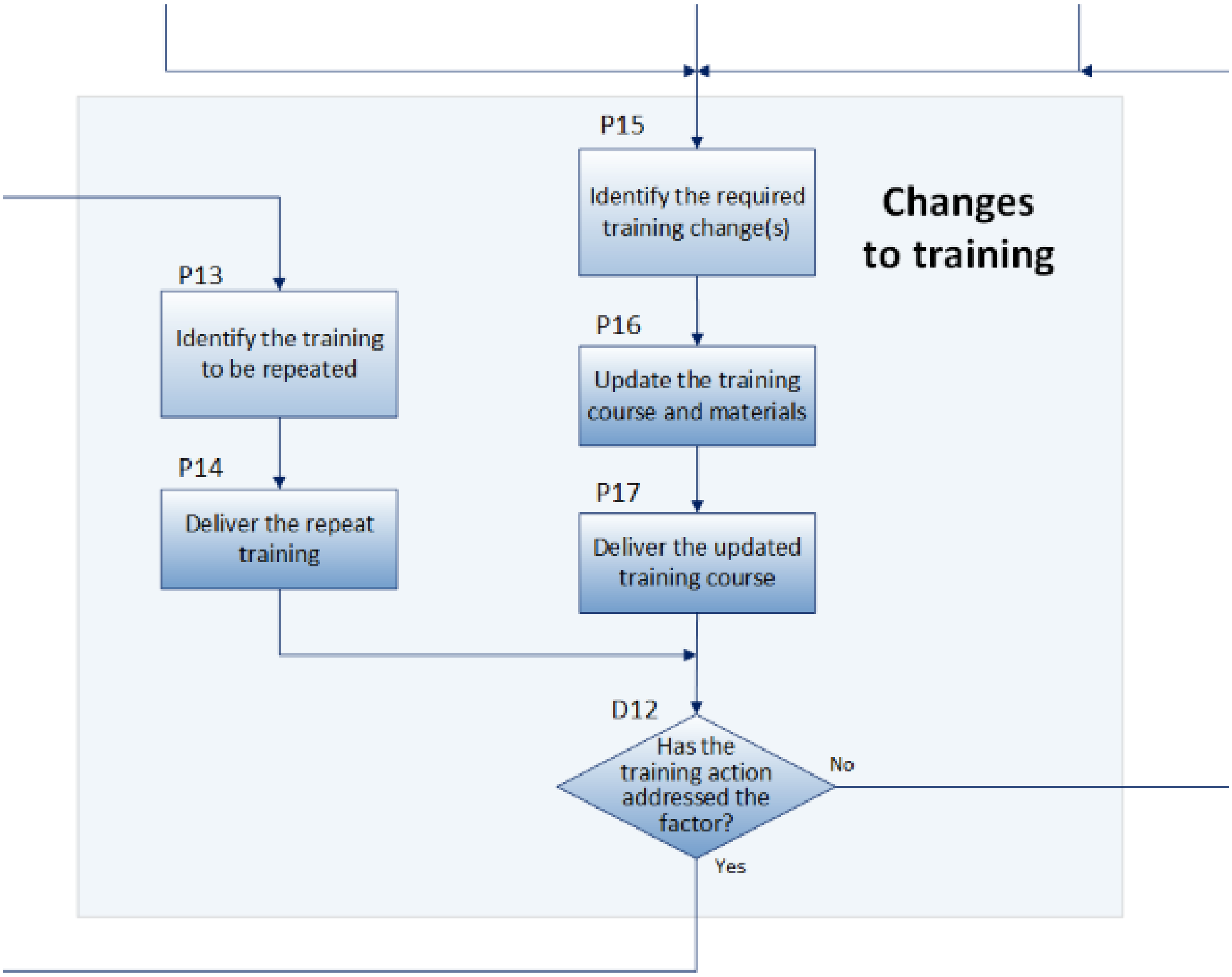
The input and trigger to the follow-up phase, changes to training, is an output from the ISHPO preparation and analysis phases that identifies whether:

- changes are required to the existing training material
- new training material is required
- repeat training is required

4.1.2 Process

The follow-up phase, changes to training (refer to Fig 2), will guide the analyst through the following steps:

1. Update the existing training course and materials (refer to Para 4.2).
2. Deliver new training (refer to Para 4.3).
3. Repeat existing training (refer to Para 4.4).
4. Assess the changes and make recommendations (refer to Para 4.5).



ICN-S6000T-B6865-00090-001-01

Fig 2 ISHPO Changes to training

4.1.3 Output

The output of the follow-up phase will be an analysis and decision that states whether the contributing factor and any Product issue has been addressed by the training changes or whether additional Product changes will be required (refer to Chap 5.2).

4.2 Identify changes and update training (P15 & P16)

Once it has been determined that a change is required in the existing training or new training is required, the updated Training Needs Analysis (TNA), must be examined. The designer should identify the Learning Objectives (LO) that were updated or added during the ISHPO analysis phase (refer to Chap 5.3) and identify which specific courses and lessons are affected by these updated or added LO. The designer should then compare the latest LO to the existing training course(s) to determine what updates are required to the syllabus, lessons plans and training materials (P15).

The instructional designer should update the syllabus, lesson plans and training materials to reflect the changes in the LO (P16).

Note

This can require new methods and media to be adopted if the TNA updates require this.

4.2.1 Output

The output of this phase will be an updated training course and associated training materials.

4.3 Deliver updated training (P17)

Once the update of the training materials has been completed, the new/revised training should be delivered to the appropriate locations and trainees (P17). During the training delivery, it is important to capture instructor and trainee feedback to:

- ensure the updated training meets the LO
- analyze the new training to ensure it has fully addressed the Product issue and contributing factor identified in the ISHPO preparation phase (refer to Para 4.5)

Note

The original training material should be saved and archived for future reference.

4.4 Repeat existing training (P13 & P14)

In many cases, especially in significant Knowledge, Skill or Attitude (KSA) decay (refer to Chap 5.3), new training and materials are not always required. Instead, the existing training could be repeated, or the course structure updated. The curriculum outline should be reviewed and updated (P13) so that the identified training is delivered at a more appropriate time or repeated during the length of the course (P14).

The training material can be made available to the operator/maintainer for future training reference or for follow-up training. However, it is important that all training materials are marked as "For Training Only", to ensure that they are not used as an operational or maintenance task reference.

4.5 Assessment and recommendations (D12)

4.5.1 Assessment

After the required changes have been completed and the updated training has been delivered, all feedback should be reviewed and analyzed to ensure the changes addressed the original contributing factor identified in the ISHPO preparation phase (refer to Chap 5.2).

The analysis should look at a variety of sources including, but not limited to:

- instructor feedback:
  - whether the changes were achievable
  - Whether the changes caused any problems or issues with the training delivery
  - how the trainees received and reacted to the training
- trainee feedback
  - formative (in-class) feedback
  - summative feedback, at the end of training
- workplace feedback
  - from users and maintainers
  - from leadership/management

All feedback collated should be used to address the question:

Have the training changes and activities addressed the contributing factor identified in the ISHPO preparation phase? (D12).

This question ultimately looks at the entire ISHPO process and identifies whether the ISHPO process has effectively addressed the factor causing the issue with the Product, so that the user and maintainer can do their tasks more easily and effectively.

4.5.2 Recommendations

At the end of the ISHPO process, the training analysts, instructional designers, and instructors need to make a recommendation about the result achieved by the ISHPO process.

If the ISHPO process has completely addressed the factor causing the issue, the analysis and design team should recommend that the process is complete and that no further training action is required (E1).

If the ISHPO process has not completely addressed the factor causing the issue, the analysis and design team should recommend that the ISHPO process go back to the Initial cause analysis phase (D1) to find an alternative solution (refer to Chap 5.2).

Chapter 6

Relationship to other specifications

Table of contents

Chapter 6.1

Relationship to other specifications - S1000D

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
S1000D	International specification for technical publications using a common source database
S1000X	S1000D input data specification

1 Introduction to S1000D

S1000D is the international specification for production and distribution of technical documentation and learning content. This specification can be applied to the documentation of any type of Product both military and civil.

Note

Since 2007, the S1000D specification has been jointly developed, maintained, and promoted by the AeroSpace and Defence Industries Association of Europe (ASD), the Aerospace Industries Association (AIA) of America, and the Airlines for America (A4A) formerly known as the Air Transport Association of America (ATA).

S1000D provides guidelines and XML schemas that support the production and distribution of a set of different types of publications and documents, using data modules, which are defined as the smallest self-contained information unit within a technical publication. Refer to S1000D.

2 Relationship between S6000T and S1000D

There are multiple touchpoints between S6000T and S1000D, both from a technical authoring perspective but also from a training material development perspective.

2.1 Training decisions impact on technical documentation

There is a set of analysis activities defined in S6000T that should influence the production of an associated technical documentation. Examples of considerations that should be taken include, but not be limited to:

- decisions on training level should impact the level of detail that is provided in eg, maintenance procedures
- decisions instructional strategy could impact the format to which its corresponding technical documentation is written
- description of target audience and their background and preferences should influence eg, authoring style and terminology to be used

2.2 Use of S1000D in the S6000T process

Development and distribution of training material is part of the overall S6000T Analysis, Design, Development, Introduction and Evaluation (ADDIE) process. However, S6000T will not in itself define any specific formats to be used for the development of training material, including development of interactive multimedia instruction using, for example, a Learning Management System (LMS).

Here can data modules as defined in S1000D support the S6000T process. Examples of data modules that can support the development of training material include, but is not limited to:

- the SCORM Content Package Module SCPM), which can be used to define the hierarchical structure of learning events within a course or learning product
- the SCO content data module which is used to define the aggregation of training information and provides references to external SCO or S1000D assets, which can define a hierarchy of training steps and can be defined as references to one or more data modules or to portions of data module content
- content data modules (eg, procedural data modules), which can be used to develop dedicated training assets that could be used as part of interactive learning (ie, be referenced by SCO content data modules)

2.3 Use of S1000D to publish and distribute results from S6000T analysis and design activities

S1000D can be used to distribute results from the S6000T analysis and design activities such as learning gaps, learning objectives and course layouts. The S1000D Schema that supports this is the learning Schema and its five different branches:

- learning plan
- learning overview
- learning content
- learning summary
- learning assessment

More details on how the S1000D learning Schema is related to the S6000T data model is described in S1000X.

2.4 Use existing S1000D publications in training development

Where applicable, existing S1000D data modules, or parts thereof (refer to S1000D) can be used as assets in training development. Refer to Para 2.2.

Chapter 6.2

Relationship to other specifications - S2000M

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
S2000M	International specification for material management
S2000X	Input data specification for S2000M
S6000X	Input data specification for S6000T

1 Introduction to 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 can 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 requirement)
- procurement planning
- order administration
- invoicing
- repair administration

S2000M is the major input to Illustrated Parts Data (IPD) data modules as defined in S1000D. Refer to S2000M.

2 Relationship between S6000T and S2000M

The relationship between S6000T and S2000M is not that extensive. There is just one touchpoint and that is the need to also take material required for training into consideration of the overall provisioning for a Product.

This can include Product related spares, consumables and support equipment which are needed specifically for training purposes, and which could need to be distributed across a set of training facilities. It can also include parts that are unique to training needed to support a specific instructional strategy including (eg, part task trainers and simulators).

Note

The acquisition of unique training equipment needs to undergo the same development and support analysis activities as any other Product.

The input to S2000M needs to clearly identify resources needed for training, both on the job or in a classroom environment. Regardless of training strategy, the tasks to be trained must be defined and the required resources must be identified together with the frequency of the need for training. A task to be trained can either be the same task as is defined during LSA or be a task that is developed for training purposes.

Details on how S2000M provisioning projects and parts data can be populated, based on data delivered in accordance with the S6000T data model, are described in S2000X.

Chapter 6.3

Relationship to other specifications - S3000L

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
S3000L	International procedure specification for Logistic Support Analysis (LSA)
S6000X	Input data specification for S6000T

1 Introduction

S3000L provides a process that ensures that logistics requirements are considered during the design of the Product and its support system. The process includes several analysis activities concerning a wide range of technical and logistics considerations and documenting the results of these activities. The achievement of an effective support system is of crucial importance concerning operation and life cycle costs. Early consideration of logistics aspects is increasingly important regarding both operational and economic aspects. A Product that cannot be operated and maintained properly and cost effectively is not acceptable to the operator. Refer to S3000L.

2 Relationship between S6000T and S3000L

2.1 General

There are multiple touchpoints between S6000T and S3000L, both from a data and from a process perspective. Training analysis and design as described in S6000T, and Logistics Support Analysis (LSA) as described in S3000L must be seen as two integrated processes where data originating from S3000L is the authoritative source for all training analysis and design activities required for jobs associated with Product support.

Where training analysis and design activities indicate that decisions made during LSA must be revised so that training can be successfully implemented, this must be fed back into the LSA process as early as possible. This also emphasizes the importance of training analysis and design activities being performed as an integral part of the early support analysis activities, including LSA.

2.2 Product breakdown structure

The first input from S3000L to S6000T is the definition of the Product and its breakdown structure. It is important that there is a common understanding on how the Product is organized into, for example, systems and subsystems, and how hardware and software parts are used in the assembly of the Product. Using the Product breakdown structure defined using S3000L, and applied also to training analysis and design, will establish the bases for a digital thread across the two support elements.

2.3 Special events and failure detection

Special events in S3000L define incidents that can be experienced by the Product during its normal operation and can impact the operational capability of the Product. Examples of special events include, but are not limited to:

- lightning strikes
- hard landings for aircraft
- grounding for a ship

Failure mode detection in S3000L identifies measurable or visible parameters whose appearance can be related, directly or indirectly, to the occurrence of an associated failure.

The outcome from these analysis activities in S3000L can also assist in identifying important tasks that the user must perform to operate the Product during a special event.

2.4 Task and subtask information

The core of S3000L is the Maintenance Task Analysis (MTA) resulting in well-defined tasks and subtasks which are required to support the Product. This includes both preventive and corrective maintenance tasks as well as operational support tasks. Task and subtask information includes duration, warnings and cautions, resource requirements and acceptance parameters etc. There is a significant amount of data for each task that can support the different analysis activities defined in S6000T including, but not limited to:

- definition of performance objectives
- identification of required training levels
- identification of required levels of learning for the tasks to be performed

Task information from S3000L should be used whenever possible during the S6000T analysis phase.

2.5 Task usage information

Task usage information in S3000L allows for one and the same task to be associated with:

- many task targets eg, parts in different modifications/configurations
- many operational and support contexts

These usage parameters can affect, for example, thresholds (intervals) that define when a task must be performed as well as the resources that can be expected to be available at different locations.

Information associated with thresholds and the availability of task resources can assist in the definition of the curriculum outline to more effectively schedule training to meet readiness requirements.

2.6 S6000X

Details on how data originating from S3000L can support the respective training analysis and design activity is described in detail in S6000X.

Chapter 6.4

Relationship to other specifications - S4000P

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
S3000L	International procedure specification for Logistic Support Analysis (LSA)
S4000P	International specification for developing and continuously improving preventive maintenance

1 Introduction to S4000P

S4000P provides methodologies to develop Preventive Maintenance Task Requirements with intervals (PMTRI). This is the authoritative source for the determination of preventive maintenance tasks that must be adhered to by operators and manufactures. Preventive maintenance task requirements and their corresponding thresholds (ie, intervals, triggers) will be defined in collaboration with specialists from operators, manufacturers, and regulatory authorities.

Specifically, S4000P outlines the general organization and decision processes for determining PMTRI initially projected for the life of the Product as well as how to continuously improve preventive maintenance throughout the in-service life of a Product, covered by the In-Service Maintenance Optimization (ISMO) process. Refer to S4000P.

2 Relationship between S6000T and S4000P

There is no direct relationship between S6000T and S4000P from a pure data perspective. All PMTRI’s identified during the preventive maintenance analysis are analyzed in detail in the Maintenance Task Analysis (MTA), defined in S3000L.

However, an early understanding of the requirements for preventive maintenance related to the Product can always inform the training analysis activities on the extent of training that can be required.

Chapter 6.5

Relationship to other specifications - S5000F

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 5	In-Service Human Performance Optimization
S5000F	International specification for in-service data feedback
S6000X	Input data specification for S6000T

1 Introduction to S5000F

S5000F provides a means for the feedback of in-service data collected and prepared by the customer to other parties to increase fleet availability and optimize Product support cost. Refer to S5000F.

2 Relationship between S6000T and S5000F

The training outline developed by the S6000T training analysis and design process must be evaluated regularly to ensure the identification of required improvements. During the in-service phase, it becomes clear if the defined training system meets the defined requirements from both the Product as well as from the trained target audience perspective.

In-service data must be collected and evaluated against the defined training system to optimize human performance, the performance of the training system, and the Product. The In-Service Human Performance Optimization (ISHPO) process defined in S6000T is the recommended process to structure the optimization, refer to Chap 5.

Details on which S5000F in-service feedback data can affect data defined during the S6000T processes is described in S6000X.

Chapter 6.6

Relationship to other specifications - SX000i

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
SX000i	International specification for Integrated Product Support (IPS)

1 Introduction

SX000i provides a guide for the use of the S-Series Integrated Product Support (IPS) specifications by IPS managers and practitioners, and provides guidance on the policy, implementation, and responsibilities for the application of the S-Series IPS specifications. It establishes the essential information required to initiate, maintain, and schedule the IPS activities that apply through the entire Product life cycle, as well as the necessary deliverables. Refer to SX000i.

2 Relationship between S6000T and SX000i

The training analysis and design process described in S6000T, is one of the central elements in the development of an efficient support solution for a Product. SX000i puts S6000T into the bigger picture of IPS and describes how training analysis and design must interact with other IPS elements in the context of the overall IPS management process covering the entire lifetime of a Product.

Chapter 7

Data model

Table of contents

Chap No.	Title		Applic
Chap 7.3.18	Units of functionality - UoF Message	S6000T-A-07-03-1800-00A-040A-A	All
Chap 7.3.19	Units of functionality - UoF Mission Definition	S6000T-A-07-03-1900-00A-040A-A	All
Chap 7.3.20	Units of functionality - UoF Organization	S6000T-A-07-03-2000-00A-040A-A	All
Chap 7.3.21	Units of functionality - UoF Part Definition	S6000T-A-07-03-2100-00A-040A-A	All
Chap 7.3.22	Units of functionality - UoF Performance Parameter	S6000T-A-07-03-2200-00A-040A-A	All
Chap 7.3.23	Units of functionality - UoF Product	S6000T-A-07-03-2300-00A-040A-A	All
Chap 7.3.24	Units of functionality - UoF Product Usage Context	S6000T-A-07-03-2400-00A-040A-A	All
Chap 7.3.25	Units of functionality - UoF Product Usage Phase	S6000T-A-07-03-2500-00A-040A-A	All
Chap 7.3.26	Units of functionality - UoF Remark	S6000T-A-07-03-2600-00A-040A-A	All
Chap 7.3.27	Units of functionality - UoF Resource Specification	S6000T-A-07-03-2700-00A-040A-A	All
Chap 7.3.28	Units of functionality - UoF Security Classification	S6000T-A-07-03-2800-00A-040A-A	All
Chap 7.3.29	Units of functionality - UoF Software Element	S6000T-A-07-03-2900-00A-040A-A	All
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization	S6000T-A-07-03-3000-00A-040A-A	All
Chap 7.3.31	Units of functionality - UoF Target Audience	S6000T-A-07-03-3100-00A-040A-A	All
Chap 7.3.32	Units of functionality - UoF Task	S6000T-A-07-03-3200-00A-040A-A	All
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude	S6000T-A-07-03-3300-00A-040A-A	All
Chap 7.3.34	Units of functionality - UoF Task Performance Objective	S6000T-A-07-03-3400-00A-040A-A	All
Chap 7.3.35	Units of functionality - UoF Task Resource	S6000T-A-07-03-3500-00A-040A-A	All
Chap 7.3.36	Units of functionality - UoF Task Target	S6000T-A-07-03-3600-00A-040A-A	All
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization	S6000T-A-07-03-3700-00A-040A-A	All
Chap 7.3.38	Units of functionality - UoF Training Analysis and Design Message Content	S6000T-A-07-03-3800-00A-040A-A	All
Chap 7.3.39	Units of functionality - UoF Training Entry Requirement	S6000T-A-07-03-3900-00A-040A-A	All

Chap No.	Title		Applic
Chap 7.3.40	Units of functionality - UoF Training Gap	S6000T-A-07-03-4000-00A-040A-A	All
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity	S6000T-A-07-03-4100-00A-040A-A	All
Chap 7.3.42	Units of functionality - UoF Training Media Resource	S6000T-A-07-03-4200-00A-040A-A	All
Chap 7.3.43	Units of functionality - UoF Training method	S6000T-A-07-03-4300-00A-040A-A	All
Chap 7.3.44	Units of functionality - UoF Training System Resource Requirement	S6000T-A-07-03-4400-00A-040A-A	All
Chap 7.3.45	Units of functionality - UoF Warning Caution Train Prioritization	S6000T-A-07-03-4500-00A-040A-A	All
Chap 7.3.46	Units of functionality - UoF Zone Element	S6000T-A-07-03-4600-00A-040A-A	All

Chapter 7.1

Data model - General

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 3	Analysis
Chap 4	Design
S3000L	International procedure specification for Logistics Support Analysis (LSA)
SX002D	Common data model for the S-Series IPS specifications
SX004G	Unified Modeling Language (UML) model readers' guidance
www.uml.org	Unified Modeling Language (UML 2.0™)

1 Introduction

This chapter defines the data that is identified during the training analysis and design activities as described in Chap 3 and Chap 4.

The S6000T data model is described using the UML 2.0™ (Unified Modeling Language) class model diagrams (www.uml.org). For guidance on how to read a UML class model refer to SX004G.

The S6000T data model is an extension of the Data Model and Exchange Working Group (DMEWG) Common Data Model issue 2.1 (refer to SX002D). This means that the S6000T data model is based on the same underlying data model as S3000L, which is an important aspect to enable a fully integrated Product support process.

Each class and attribute defined in the S6000T data model is defined as part of the Unit of Functionality (UoF) where it is first encountered.

2 Scope

The following areas are within scope of the S6000T issue 2.1 data model:

- Product structure to be supported
- tasks including tasks where preceding Logistics Support Analysis (LSA) activities are the authoritative source for tasks to be considered for training
- identification of missions to be carried out by the Product and its need for human interaction
- identification of jobs, duties, and tasks
- definition of performance objectives
- prioritization of tasks, subtasks, and subtask steps to be trained
- definition of required Knowledge, Skills, and Attitudes (KSA) for the respective task, subtask and subtask step that has been prioritized for training
- description of the training target audience and their occupational backgrounds
- identification of KSA gaps for the training target audience
- definition of learning objectives, learning assessment strategies and instructional strategies
- layout for curriculums, courses, and course content

3 Out of scope

The data model for S6000T does not cover all the aspects that need to be documented as part of a detailed task analysis but only those aspects that are important for making decisions related to training. Refer to S3000L.

Chapter 7.2

Data model - Overview

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3	Data model - Units of functionality
Chap 7.3.1	Units of functionality - UoF Aggregated element
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.5	Units of functionality - UoF Circuit Breaker
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.12	Units of functionality - UoF Environment Definition
Chap 7.3.13	Units of functionality - UoF Hardware Element
Chap 7.3.14	Units of functionality - UoF Job Duty

Chap No./Document No.	Title
Chap 7.3.15	Units of functionality - UoF Learning Assessment
Chap 7.3.16	Units of functionality - UoF Learning Gap
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.18	Units of functionality - UoF Message
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.20	Units of functionality - UoF Organization
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.22	Units of functionality - UoF Performance Parameter
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.24	Units of functionality - UoF Product Usage Context
Chap 7.3.25	Units of functionality - UoF Product Usage Phase
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.29	Units of functionality - UoF Software Element
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
Chap 7.3.34	Units of functionality - UoF Task Performance Objective
Chap 7.3.35	Units of functionality - UoF Task Resource
Chap 7.3.36	Units of functionality - UoF Task Target
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization
Chap 7.3.38	Units of functionality - UoF Training Analysis and Design Message Content
Chap 7.3.39	Units of functionality - UoF Training Entry Requirement
Chap 7.3.40	Units of functionality - UoF Training Gap
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity
Chap 7.3.42	Units of functionality - UoF Training Media Resource
Chap 7.3.43	Units of functionality - UoF Training Method
Chap 7.3.44	Units of functionality - UoF Training System Resource Requirement
Chap 7.3.45	Units of functionality - UoF Warning Caution Train Prioritization
Chap 7.3.46	Units of functionality - UoF Zone Element

Chap No./Document No.	Title
S3000L	International procedure specification for Logistics Support Analysis (LSA)

1 Data model overview

1.1 S6000T UoF overview

The S6000T data model is organized into a set of 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 first set of UoF, documents the Product to be supported including its breakdown structure and parts. This is data that must be common to all Product support activities and not be data that is defined for S6000T alone. This includes:

- UoF Product (refer to Chap 7.3.23)
- UoF Product usage phase (refer to Chap 7.3.25)
- UoF Breakdown structure (refer to Chap 7.3.3) and its specializations for:
  - UoF Hardware element (refer to Chap 7.3.13)
  - UoF Software element (refer to Chap 7.3.29)
  - UoF Aggregated element (refer to Chap 7.3.1)
  - UoF Zone element (refer to Chap 7.3.46)
- UoF Part definition (refer to Chap 7.3.21)

The second set of UoF, documents task information independent of whether a task has been identified as part of a Logistic Support Analysis (LSA) (eg, S3000L) or if it has been identified during the initial training analysis activities. Specifically, tasks coming out from LSA must be common to all Product support activities and not be data that is defined for S6000T alone. This includes:

- UoF Task (refer to Chap 7.3.32)
- UoF Circuit breaker (refer to Chap 7.3.5)
- UoF Task resource (refer to Chap 7.3.35)
- UoF Resource specification (refer to Chap 7.3.27)
- UoF Task target (refer to Chap 7.3.36)

The third set of UoF, defines data which are unique to training analysis. This includes:

- UoF Mission definition (refer to Chap 7.3.19)
- UoF Performance parameter (refer to Chap 7.3.22)
- UoF Job duty (refer to Chap 7.3.14)
- UoF Task performance objective (refer to Chap 7.3.34)
- UoF Task train prioritization (refer to Chap 7.3.37)
- UoF Subtask train prioritization (refer to Chap 7.3.30)
- UoF Warning caution train prioritization (refer to Chap 7.3.45)
- UoF Task knowledge skill and attitude (refer to Chap 7.3.33)
- UoF Training Gap (refer to Chap 7.3.40)

The fourth set of UoF, defines data which are unique to training design. This includes:

- UoF Training entry requirement (refer to Chap 7.3.39)
- UoF Target audience (refer to Chap 7.3.31)
- UoF Competency definition (refer to Chap 7.3.6)
- UoF Learning gap (refer to Chap 7.3.16)
- UoF Learning objective (refer to Chap 7.3.17)

- UoF Learning assessment (refer to Chap 7.3.15)
- UoF Training method (refer to Chap 7.3.43)
- UoF Training media and fidelity (refer to Chap 7.3.41)
- UoF Training media resource (refer to Chap 7.3.42)
- UoF Curriculum and course plan (refer to Chap 7.3.9)
- UoF Course definition (refer to Chap 7.3.7)
- UoF Course element (refer to Chap 7.3.8)
- UoF Training system resource requirement (refer to Chap 7.3.44)

There is also a set of supporting UoF which allows for additional information to be associated with business objects (UML classes) defined in the UoF defined above. This includes:

- UoF Applicability statement (refer to Chap 7.3.2)
- UoF Change information (refer to Chap 7.3.4)
- UoF Digital file (refer to Chap 7.3.10)
- UoF Document (refer to Chap 7.3.11)
- UoF Organization (refer to Chap 7.3.20)
- UoF Remark (refer to Chap 7.3.26)
- UoF Security classification (refer to Chap 7.3.28)
- UoF Environment definition (refer to Chap 7.3.12)
- UoF Product usage context (refer to Chap 7.3.24)

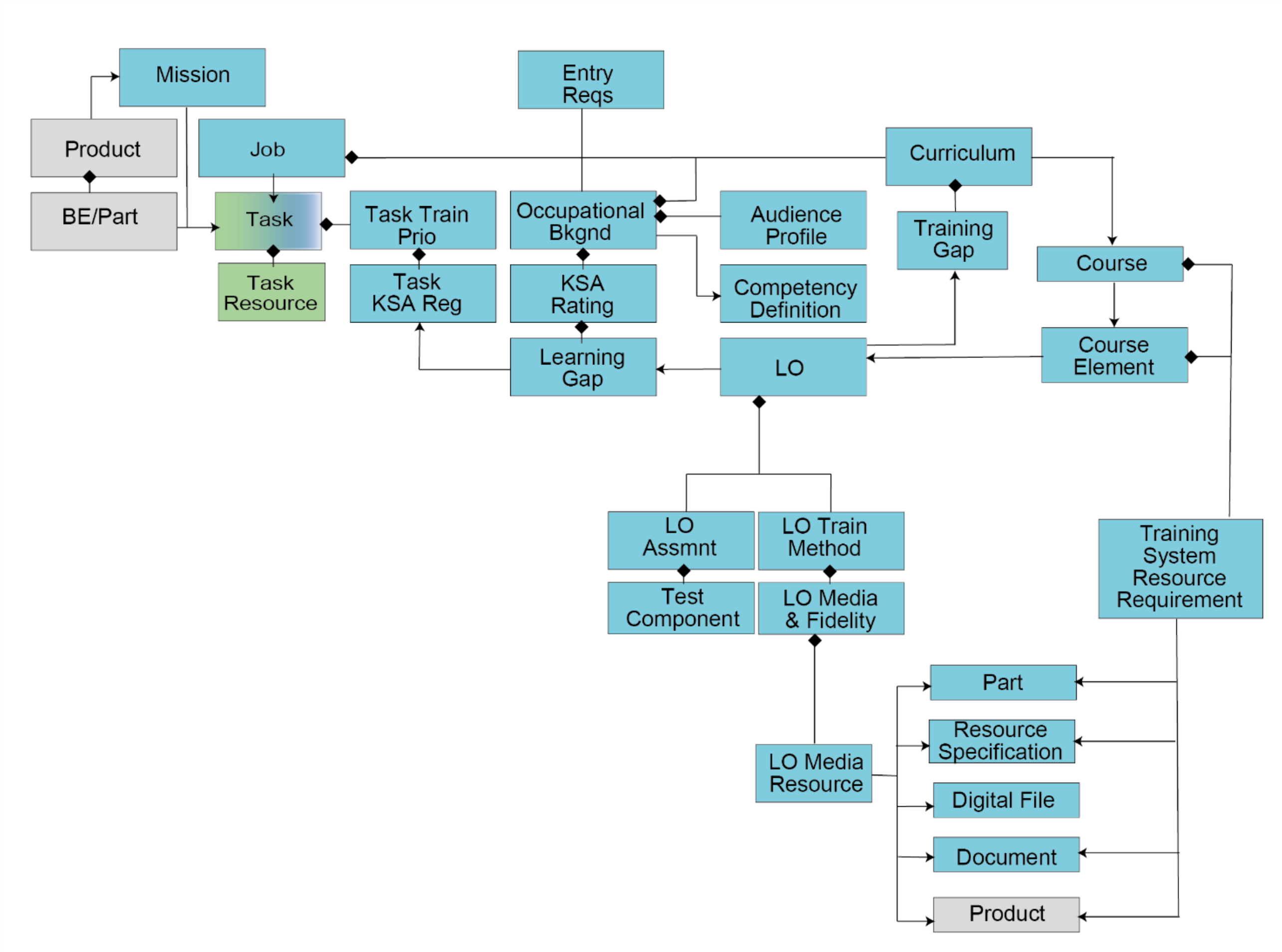
The last set of UoF defines the constructs needed to exchange data between stakeholders of data resulting from the training analysis and design activities. This includes:

- UoF Message (refer to Chap 7.3.18)
- UoF Training analysis and design message content (refer to Chap 7.3.38)

2 S6000T Business objects overview

The S6000T data model is made up from a set of UML classes (business objects), each representing an important set of information important to training analysis and design.

A high-level business object model is illustrated in Fig 1.



ICN-S6000T-B6865-00026-003-01

Fig 1 S6000T business objects overview

Business objects marked in grey represent information that typically originates from the Product design and development process, and which could have been further refined during the initial Product support analysis activities (eg, LSA). The types of business objects that originate from Product design and development includes information about the Product itself and its variants, hardware and software parts and allowed configurations.

Business objects marked in green represent information that typically originates from the initial Product support analysis activities and primarily LSA. This includes information about Product support tasks and its subtasks, warnings, cautions, and resource requirements.

Business objects marked in blue represent information that is developed during training analysis and training design and is the authoritative source for further development of training material and training intervention.

Chapter 7.3

Data model - Units of functionality

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.1	Units of functionality - UoF Aggregated Element
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.5	Units of functionality - UoF Circuit Breaker
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.12	Units of functionality - UoF Environment Definition
Chap 7.3.13	Units of functionality - UoF Hardware Element
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.15	Units of functionality - UoF Learning Assessment
Chap 7.3.16	Units of functionality - UoF Learning Gap
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.18	Units of functionality - UoF Message
Chap 7.3.19	Units of functionality - UoF Mission Definition

Chap No./Document No.	Title
Chap 7.3.20	Units of functionality - UoF Organization
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.22	Units of functionality - UoF Performance Parameter
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.24	Units of functionality - UoF Product Usage Context
Chap 7.3.25	Units of functionality - UoF Product Usage Phase
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.29	Units of functionality - UoF Software Element
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
Chap 7.3.34	Units of functionality - UoF Task Performance Objective
Chap 7.3.35	Units of functionality - UoF Task Resource
Chap 7.3.36	Units of functionality - UoF Task Target
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization
Chap 7.3.38	Units of functionality - UoF Training Analysis and Design Message Content
Chap 7.3.39	Units of functionality - UoF Training Entry Requirement
Chap 7.3.40	Units of functionality - UoF Training Gap
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity
Chap 7.3.42	Units of functionality - UoF Training Media Resource
Chap 7.3.43	Units of functionality - UoF Training Method
Chap 7.3.44	Units of functionality - UoF Training System Resource Requirement
Chap 7.3.45	Units of functionality - UoF Warning Caution Train Prioritization
Chap 7.3.46	Units of functionality - UoF Zone Element

1 General

The S6000T Data Model is organized into a set of Units of Functionality (UoF). Each UoF defines a set of business objects and data elements that together support the definition of data for a particular subject.

A UoF can refer to business objects in other UoF to bring the referred data into larger contexts.

Splitting the S6000T data model into a set of UoF provides small and coherent portions of the data model to help the reader understand the complete data model.

2 Units of functionality - Definitions

The definitions of the UoF are given in the chapters listed in Table 1.

Chapter 7.3.1

Units of functionality - UoF Aggregated Element

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Hardware element
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.46	Units of functionality - UoF Zone Element
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

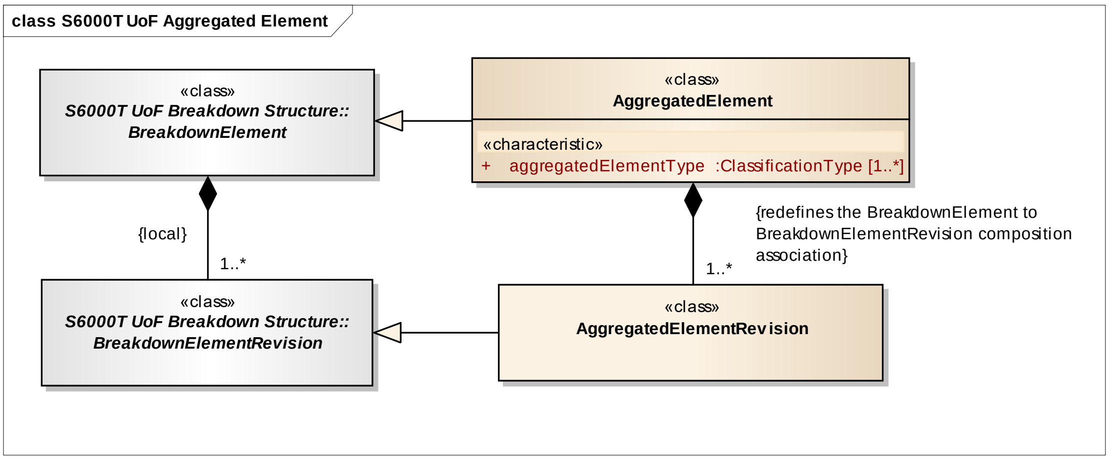
1 General

The Aggregated Element UoF provides the capability to specify that an element within a breakdown represents a collection of elements for an identified purpose.

2 UoF Aggregated Element

Key features of the UoF Aggregated Element data model, (refer to Fig 1), are:

- AggregatedElement is a specialization of BreakdownElement, which means that wherever BreakdownElement is being used in the data model AggregatedElement can be used instead
- AggregatedElementRevision is a specialization of BreakdownElementRevision, which means that wherever BreakdownElementRevision is being used in the data model AggregatedElementRevision can be used instead



ICN-S6000T-B6865-00032-002-01

Fig 1 UoF Aggregated Element

2.1 AggregatedElement

2.1.1 Class definition

AggregatedElement is a BreakdownElement (refer to Chap 7.3.3) that is a container for a collection of BreakdownElements which are grouped for an identified purpose.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of AggregatedElementRevision

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem (inherited from BreakdownElement). Refer to Chap 7.3.46.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

- SecurityClassificationItem (inherited from BreakdownElement). Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 breakdownElementIdentifier (one or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.1.2.2 breakdownElementName (zero, one or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.1.2.3 breakdownElementEssentiality (zero or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.1.2.4 aggregatedElementType (one or many)
aggregatedElementType is a classification that identifies further specialization for an AggregatedElement.

Source:

- SX001G

Valid values:

- FA (SX001G:familyBreakdownElement)
- FU (SX001G:functionBreakdownElement)
- GR (SX001G:groupBreakdownElement)
- SY (SX001G:systemBreakdownElement)

2.2 AggregatedElementRevision

2.2.1 Class definition

AggregatedElementRevision is a BreakdownElementRevision (refer to Chap 7.3.3) representing an iteration applied to an AggregatedElement.

Source:

- SX001G

2.2.1.1 Implementations
This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem (inherited from BreakdownElementRevision). Refer to Chap 7.3.46.
- ChangeControlledItem (inherited from BreakdownElementRevision). Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 breakdownElementRevisionIdentifier
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.2.2.2 breakdownElementDescription (zero, one or many)
Inherited from `BreakdownElementRevision`. Refer to Chap 7.3.3.

2.2.2.3 breakdownElementRevisionRationale (zero, one or many)
Inherited from `BreakdownElementRevision`. Refer to Chap 7.3.3.

2.2.2.4 breakdownElementRevisionDate (zero or one)
Inherited from `BreakdownElementRevision`. Refer to Chap 7.3.3.

2.2.2.5 breakdownElementRevisionStatus (zero or one)
Inherited from `BreakdownElementRevision`. Refer to Chap 7.3.3.

Chapter 7.3.2

Units of functionality - UoF Applicability Statement

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Hardware element
Chap 7.3.5	Units of functionality - UoF Software element
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.12	Units of functionality - UoF Environment Definition
Chap 7.3.13	Units of functionality - UoF Hardware Element
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.20	Units of functionality - UoF Organization
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.22	Units of functionality - UoF Performance Parameter
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.24	Units of functionality - UoF Product Usage Context
Chap 7.3.25	Units of functionality - UoF Product Usage Phase
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.29	Units of functionality - UoF Software Element
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization

Chap No./Document No.	Title
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
Chap 7.3.34	Units of functionality - UoF Task Performance Objective
Chap 7.3.35	Units of functionality - UoF Task Resource
Chap 7.3.36	Units of functionality - UoF Task Target
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization
Chap 7.3.42	Units of functionality - UoF Training Media Resource
Chap 7.3.43	Units of functionality - UoF Training Method
Chap 7.3.44	Units of functionality - UoF Training System Resource Requirement
Chap 7.3.45	Units of functionality - UoF Warning Caution Train Prioritization
Chap 7.3.46	Units of functionality - UoF Zone Element
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

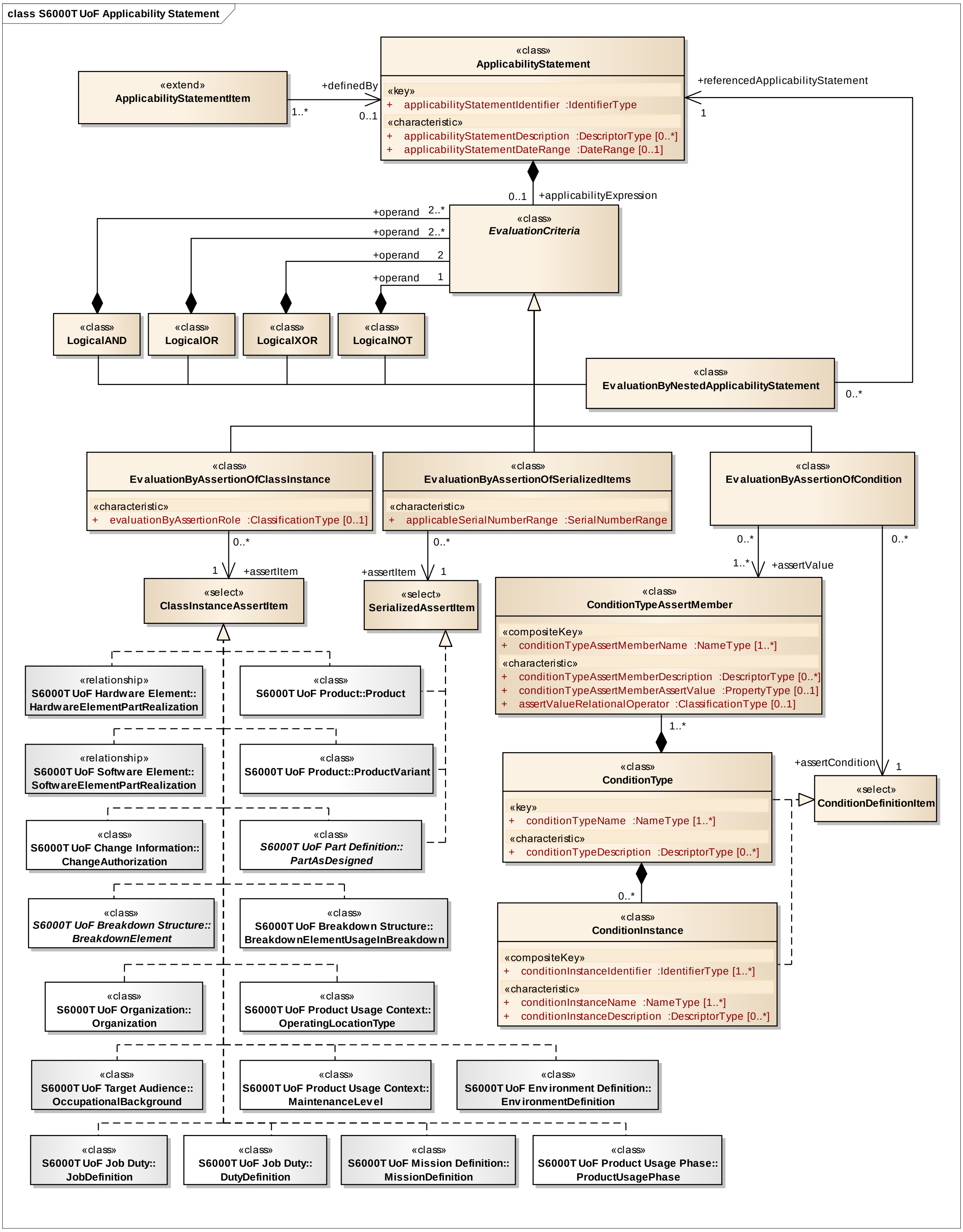
The Applicability Statement UoF provides the capability to define the situation or situations under which related items are valid.

2 UoF Applicability Statement

Key features of the UoF Applicability Statement data model (refer to Fig 1 and Fig 2) are:

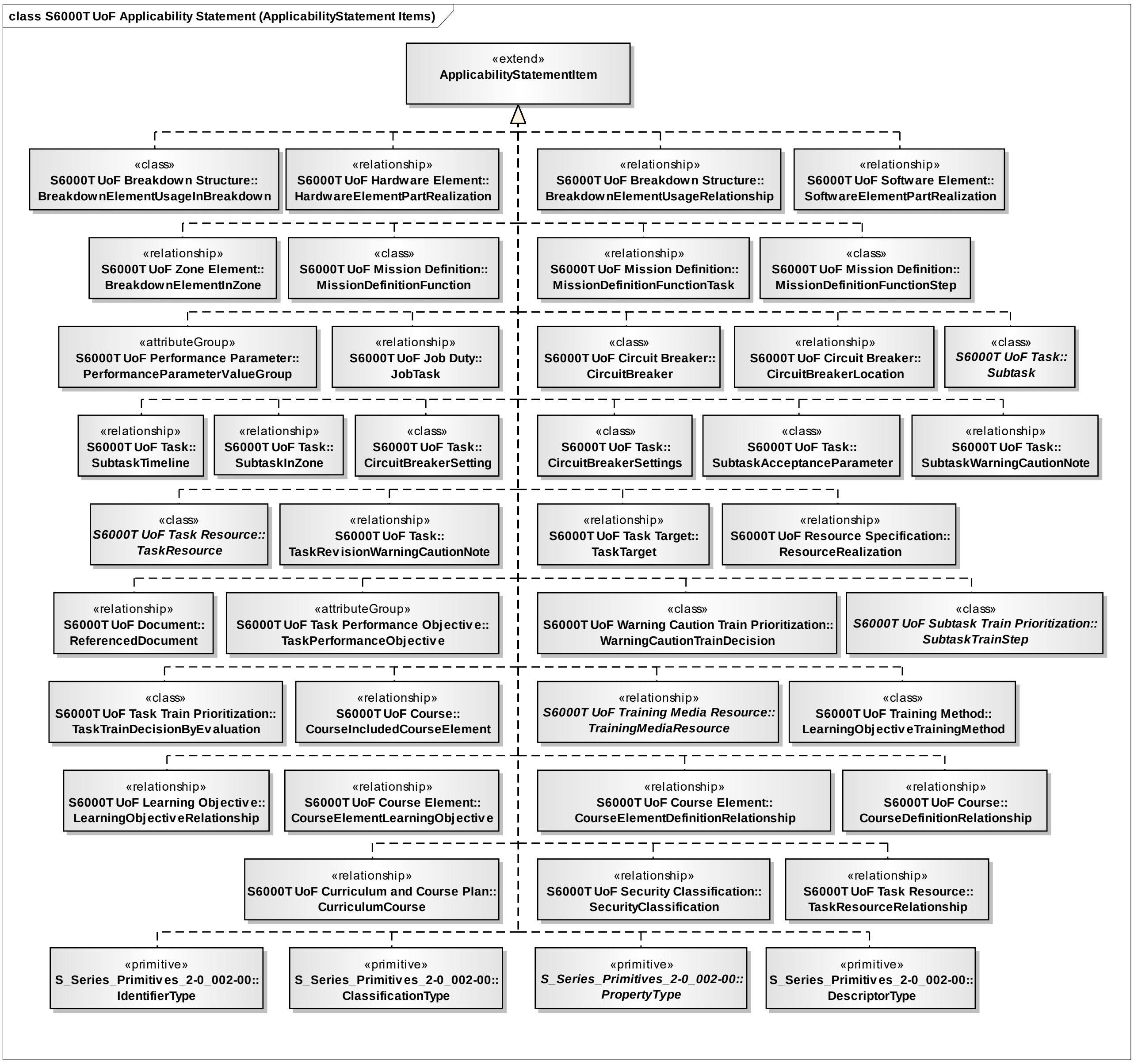
- it consists of three major components:
  - identification and definition of the applicability statement
  - definition of the evaluation expression (evaluation criteria)
  - identification of the data that is to be asserted as part of the evaluation expression
- the applicability statement does not require an evaluation expression but can contain only a textual expression
- the evaluation criteria is always computer interpretable, and it can be calculated to be TRUE or FALSE
- the evaluation criteria must be defined as either:
  - Logical AND
  - Logical OR
  - Logical XOR
  - Logical NOT
  - evaluation by nested applicability statement
  - evaluation by assertion of class instance
  - evaluation by assertion of serialized items
  - evaluation by assertion of condition

- The value to be asserted can either be an existing value defined by other classes in the data model (evaluation by assertion of class instance or evaluation by assertion of serialized items) or be an additional condition



ICN-S6000T-B6865-00065-002-01

Fig 1 UoF Applicability Statement



ICN-S6000T-B6865-00066-002-01

Fig 2 UoF Applicability Statement - Applicability Statement Items

2.1 ApplicabilityStatement

2.1.1 Class definition

`ApplicabilityStatement` is a `<<class>>` that defines the situation or situations under which related items are valid.

Source:

– SX001G

2.1.1.1 Associations

This class has the following associations:

– An aggregate association with zero or one instance of `EvaluationCriteria`

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 applicabilityStatementIdentifier

applicabilityStatementIdentifier is an identifier that establishes a unique designator for an ApplicabilityStatement and to differentiate it from other instances of ApplicabilityStatement.

Valid identifier class values:

- ID (SX001G:applicabilityStatementIdentifier)

Source:

- SX001G

2.1.2.2 applicabilityStatementDescription (zero, one or many)

applicabilityStatementDescription is a description that provides a human readable expression of the defined rule.

Source:

- SX001G

2.1.2.3 applicabilityStatementDateRange (zero or one)

applicabilityStatementDateRange is a date range that defines the date interval for when the applicability evaluation can result in a TRUE result.

Note

If outside that date range, the ApplicabilityStatement always results in a FALSE statement.

Source:

- SX001G

2.2 ApplicabilityStatementItem

2.2.1 Interface definition

ApplicabilityStatementItem is an <<extend>> interface that provides its associated data model to those classes which can have restricted validity as defined by an associated ApplicabilityStatement.

Source:

- SX001G

2.2.1.1 Associations

The ApplicabilityStatementItem <<extend>> interface has the following associations:

- A directed definedBy association with zero or one instance of ApplicabilityStatement

2.2.1.2 Class members

Classes that implement the ApplicabilityStatementItem <<extend>> interface are:

- BreakdownElementInZone. Refer to Chap 7.3.46.
- BreakdownElementUsageInBreakdown. Refer to Chap 7.3.3.
- BreakdownElementUsageRelationship. Refer to Chap 7.3.3.
- CircuitBreaker. Refer to Chap 7.3.5.
- CircuitBreakerLocation. Refer to Chap 7.3.5.
- CircuitBreakerSetting. Refer to Chap 7.3.32.
- CircuitBreakerSettings. Refer to Chap 7.3.32.
- ClassificationType. Refer to SX002D.
- CourseDefinitionRelationship. Refer to Chap 7.3.7.
- CourseElementDefinitionRelationship. Refer to Chap 7.3.8.
- CourseIncludedCourseElement. Refer to Chap 7.3.7.
- CurriculumCourse. Refer to Chap 7.3.9.
- DescriptorType. Refer to SX002D.
- HardwareElementPartRealization. Refer to Chap 7.3.13.
- IdentifierType. Refer to SX002D.
- JobTask. Refer to Chap 7.3.14.
- LearningObjectiveRelationship. Refer to Chap 7.3.17.
- LearningObjectiveTrainingMethod. Refer to Chap 7.3.43.
- MissionDefinitionFunction. Refer to Chap 7.3.19.
- MissionDefinitionFunctionStep. Refer to Chap 7.3.19.
- MissionDefinitionFunctionTask. Refer to Chap 7.3.19.
- PerformanceParameterValueGroup. Refer to Chap 7.3.22.
- PropertyType. Refer to SX002D.
- ReferencedDocument. Refer to Chap 7.3.11.
- ResourceRealization. Refer to Chap 7.3.27.
- SecurityClassification. Refer to Chap 7.3.28.
- SoftwareElementPartRealization. Refer to Chap 7.3.29.
- Subtask. Refer to Chap 7.3.32.
- SubtaskAcceptanceParameter. Refer to Chap 7.3.32.
- SubtaskInZone. Refer to Chap 7.3.32.
- SubtaskPerformanceObjective. Refer to Chap 7.3.30.
- SubtaskStepPerformanceObjective. Refer to Chap 7.3.30.
- SubtaskTimeline. Refer to Chap 7.3.32.
- SubtaskTrainStep. Refer to Chap 7.3.30.
- SubtaskWarningCautionNote. Refer to Chap 7.3.32.
- TaskPerformanceObjective. Refer to Chap 7.3.34.
- TaskResource. Refer to Chap 7.3.35.
- TaskResourceRelationship. Refer to Chap 7.3.35.
- TaskRevisionWarningCautionNote. Refer to Chap 7.3.32.
- TaskTarget. Refer to Chap 7.3.36.
- TaskTrainDecisionByEvaluation. Refer to Chap 7.3.37.
- TrainingMediaResource. Refer to Chap 7.3.42.

- TrainingSystemResourceRequirement. Refer to Chap 7.3.44.
- WarningCautionTrainDecision. Refer to Chap 7.3.45.

2.3 ClassInstanceAssertItem

2.3.1 Interface definition

ClassInstanceAssertItem is a <<select>> interface that identifies classes from which an instance can be used as the EvaluationByAssertionOfClassInstance assert item.

Source:

- SX001G

2.3.1.1 Class members

This <<select>> interface includes the following class members:

- BreakdownElement. Refer to Chap 7.3.3.
- BreakdownElementUsageInBreakdown. Refer to Chap 7.3.3.
- ChangeAuthorization. Refer to Chap 7.3.4.
- DutyDefinition. Refer to Chap 7.3.14.
- EnvironmentDefinition. Refer to Chap 7.3.12.
- HardwareElementPartRealization. Refer to Chap 7.3.13.
- JobDefinition. Refer to Chap 7.3.14.
- MaintenanceLevel. Refer to Chap 7.3.24.
- MissionDefinition. Refer to Chap 7.3.19.
- OccupationalBackground. Refer to Chap 7.3.31.
- OperatingLocationType. Refer to Chap 7.3.24.
- Organization. Refer to Chap 7.3.20.
- PartAsDesigned. Refer to Chap 7.3.21.
- Product. Refer to Chap 7.3.23.
- ProductUsagePhase. Refer to Chap 7.3.25.
- ProductVariant. Refer to Chap 7.3.23.
- SoftwareElementPartRealization. Refer to Chap 7.3.29.

2.4 ConditionDefinitionItem

2.4.1 Interface definition

ConditionDefinitionItem is a <<select>> interface that identifies classes from which an instance can be used as the EvaluationByAssertionOfCondition assert condition.

Source:

- SX001G

2.4.1.1 Class members

This <<select>> interface includes the following class members:

- ConditionInstance
- ConditionType

2.5 ConditionInstance

2.5.1 Class definition

ConditionInstance is a <<class>> that defines an individual concept or object having the characteristics of a generic ConditionType.

Example:

- uniquely identified service bulletin

Source:

- SX001G

2.5.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 conditionInstanceIdentifier (one or many)

conditionInstanceIdentifier is an identifier that establishes a unique designator for a ConditionInstance and to differentiate it from other instances of ConditionInstance.

Valid identifier class values:

- ID (SX001G:conditionInstanceIdentifier)

Source:

- SX001G

2.5.2.2 conditionInstanceName (one or many)

conditionInstanceName is a name by which the ConditionInstance is known and can be easily referenced.

Source:

- SX001G

2.5.2.3 conditionInstanceDescription (zero, one or many)

conditionInstanceDescription is a description that gives more information on the meaning of the ConditionInstance.

Source:

- SX001G

2.6 ConditionType

2.6.1 Class definition

ConditionType is a <<class>> that defines a concept or an object that needs to be included in applicability statements where the concept or object is not already represented in the data model.

Example:

- environmental conditions

Source:

- SX001G

2.6.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `ConditionInstance`
- An aggregate association with one or many instances of `ConditionTypeAssertMember`

2.6.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 conditionTypeName (one or many)

`conditionTypeName` is a name by which the `ConditionType` is known and can be easily referenced.

Examples:

- ashore or afloat
- maintenance environment
- operational environment
- service bulletin

Source:

- SX001G

2.6.2.2 conditionTypeDescription (zero, one or many)

`conditionTypeDescription` is a description that gives more information on the meaning of the `ConditionType`.

Source:

- SX001G

2.7 ConditionTypeAssertMember

2.7.1 Class definition

`ConditionTypeAssertMember` is a `<<class>>` that defines a member for a given `ConditionType` which can be mapped to a Boolean expression and be evaluated to be either `TRUE` or `FALSE`.

Source:

- SX001G

2.7.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.7.2 Attributes

2.7.2.1 conditionTypeAssertMemberName (one or many)

conditionTypeAssertMemberName is a name that identifies a ConditionTypeAssertMember.

Source:

- SX001G

2.7.2.2 conditionTypeAssertMemberDescription (zero, one or many)

conditionTypeAssertMemberDescription is a description that gives more information on meaning of the ConditionTypeAssertMember.

Source:

- SX001G

2.7.2.3 conditionTypeAssertMemberAssertValue (zero or one)

conditionTypeAssertMemberAssertValue is a numerical property that specifies values which can be used to further characterize the ConditionTypeAssertMember.

Valid unit value:

- Unit

Source:

- SX001G

2.7.2.4 assertValueRelationalOperator (zero or one)

assertValueRelationalOperator is a classification that identifies a mathematical operation to be applied when testing a value against a defined conditionTypeAssertMemberAssertValue.

Examples:

- greater than
- less than

Valid values:

- EQ (SX001G:equalToComparisonOperator)
- NE (SX001G:notEqualToComparisonOperator)
- LT (SX001G:lessThanComparisonOperator)
- LE (SX001G:lessThanOrEqualToComparisonOperator)
- GT (SX001G:greaterThanComparisonOperator)
- GE (SX001G:greaterThanOrEqualToComparisonOperator)
- IN (SX001G:withinRangeComparisonOperator)

– OUT (SX001G:outsideRangeComparisonOperator)

Source:

– SX001G

2.8 EvaluationByAssertionOfClassInstance

2.8.1 Class definition

EvaluationByAssertionOfClassInstance is an EvaluationCriteria that identifies a class instance to be used as an assert item and be mapped to a Boolean expression which can be evaluated to be either TRUE or FALSE.

Source:

– SX001G

2.8.1.1 Associations

This class has the following associations:

– A directed assertItem association with one instance of ClassInstanceAssertItem

2.8.1.2 Implementations

This class implements the following <<extend>> interfaces:

– DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
– ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.8.2 Attributes

2.8.2.1 evaluationByAssertionRole (zero or one)

evaluationByAssertionRole is a classification that defines the context in which the EvaluationByAssertionOfClassInstance is being referenced.

Valid values:

– CUS (SX001G:customer)
– OP (SX001G:operator)

Source:

– SX001G

2.9 EvaluationByAssertionOfCondition

2.9.1 Class definition

EvaluationByAssertionOfCondition is an EvaluationCriteria that identifies a combination of a defined condition and a defined value to be used as an assert item and be mapped to a Boolean expression which can be evaluated to be either TRUE or FALSE.

Source:

– SX001G

2.9.1.1 Associations

This class has the following associations:

- A directed assertValue association with one instance of ConditionTypeAssertMember
- A directed assertCondition association with one instance of ConditionDefinitionItem

2.9.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.10 EvaluationByAssertionOfSerializedItems

2.10.1 Class definition

EvaluationByAssertionOfSerializedItems is an EvaluationCriteria that identifies a class instance together with an associated serial number range to be used as an assert item and be mapped to a Boolean expression which can be evaluated to be either TRUE or FALSE

Source:

- SX001G

2.10.1.1 Associations

This class has the following associations:

- A directed assertItem association with one instance of SerializedAssertItem

2.10.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.10.2 Attributes

2.10.2.1 applicableSerialNumberRange

applicableSerialNumberRange is a serial number range that identifies a limited effectivity with respect to a given interval of serialized items.

Source:

- SX001D

2.11 EvaluationByNestedApplicabilityStatement

2.11.1 Class definition

EvaluationByNestedApplicabilityStatement is an EvaluationCriteria that enables an ApplicabilityStatement to be reused as part of this EvaluationCriteria.

Note

This class enables the definition of nested applicability statements.

Source:

- SX001G

2.11.1.1 Associations

This class has the following associations:

- A directed `referencedApplicabilityStatement` association with one instance of `ApplicabilityStatement`

2.11.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.12 EvaluationCriteria

2.12.1 Class definition

`EvaluationCriteria` is a `<<class>>` that defines conditions that can be mapped to a Boolean expression which can be evaluated to be either `TRUE` or `FALSE`.

Source:

- SX001G

2.12.1.1 Implementations

This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.13 LogicalAND

2.13.1 Class definition

`LogicalAND` is an `EvaluationCriteria` that defines a Boolean operation where the results of all its associated `EvaluationCriteria` must be `TRUE` for the result to be `TRUE`, otherwise the result is `FALSE`.

Source:

- SX001G

2.13.1.1 Associations

This class has the following associations:

- An aggregate `operand` association with two or more instances of `EvaluationCriteria`

2.13.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.14 LogicalNOT

2.14.1 Class definition

LogicalNOT is an EvaluationCriteria that defines a Boolean operation where the result from its associated EvaluationCriteria must be FALSE for the result to be TRUE, otherwise the result is FALSE.

Source:

- SX001G

2.14.1.1 Associations

This class has the following associations:

- An aggregate operand association with one instance of EvaluationCriteria

2.14.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.15 LogicalOR

2.15.1 Class definition

LogicalOR is an EvaluationCriteria that defines a Boolean operation where the result from at least one of its associated EvaluationCriteria must be TRUE for the result to be TRUE, otherwise the result is FALSE.

Source:

- SX001G

2.15.1.1 Associations

This class has the following associations:

- An aggregate operand association with two or more instances of EvaluationCriteria

2.15.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.

– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.16 LogicalXOR

2.16.1 Class definition

LogicalXOR is an EvaluationCriteria that defines a Boolean operation where the result from one and only one of its associated EvaluationCriteria must be TRUE for the result to be TRUE, otherwise the result is FALSE.

Source:

– SX001G

2.16.1.1 Associations

This class has the following associations:

– An aggregate operand association with exactly two instances of EvaluationCriteria

2.16.1.2 Implementations

This class implements the following <<extend>> interfaces:

– DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
– ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.17 SerializedAssertItem

2.17.1 Interface definition

SerializedAssertItem is a <<select>> interface that identifies classes from which an instance can be used as the EvaluationByAssertionOfSerializedItems assert item.

Source:

– SX001G

2.17.1.1 Class members

This <<select>> interface includes the following class members:

– PartAsDesigned. Refer to Chap 7.3.21.
– Product. Refer to Chap 7.3.23.
– ProductVariant. Refer to Chap 7.3.23.

Chapter 7.3.3

Units of functionality - UoF Breakdown Structure

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.4.46	Units of functionality - UoF Zone Element
S1000D	International specification for technical publications using a common source database
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications
SAE-GEIA-STD-0007	Logistics Product Data

1 General

The Breakdown Structure UoF provides the capability to define any number of hierarchical structures for a specific Product or Product variant.

2 UoF Breakdown Structure

Key features of the UoF Breakdown Structure data model, (refer to Fig 1), are:

- breakdowns can be defined either for the overall Product or per specific Product variant
- each breakdown can have one or many breakdown revisions, where each breakdown revision references the breakdown element revisions that are included in the specific revision of the breakdown
- references between a breakdown revision and its constituent breakdown elements are established using the `BreakdownElementUsageInBreakdown` class
- breakdown elements included in a breakdown revision can be organized hierarchically into a breakdown structure by associating the `BreakdownElementStructure` class to instances of `BreakdownElementUsageInBreakdown`

Note

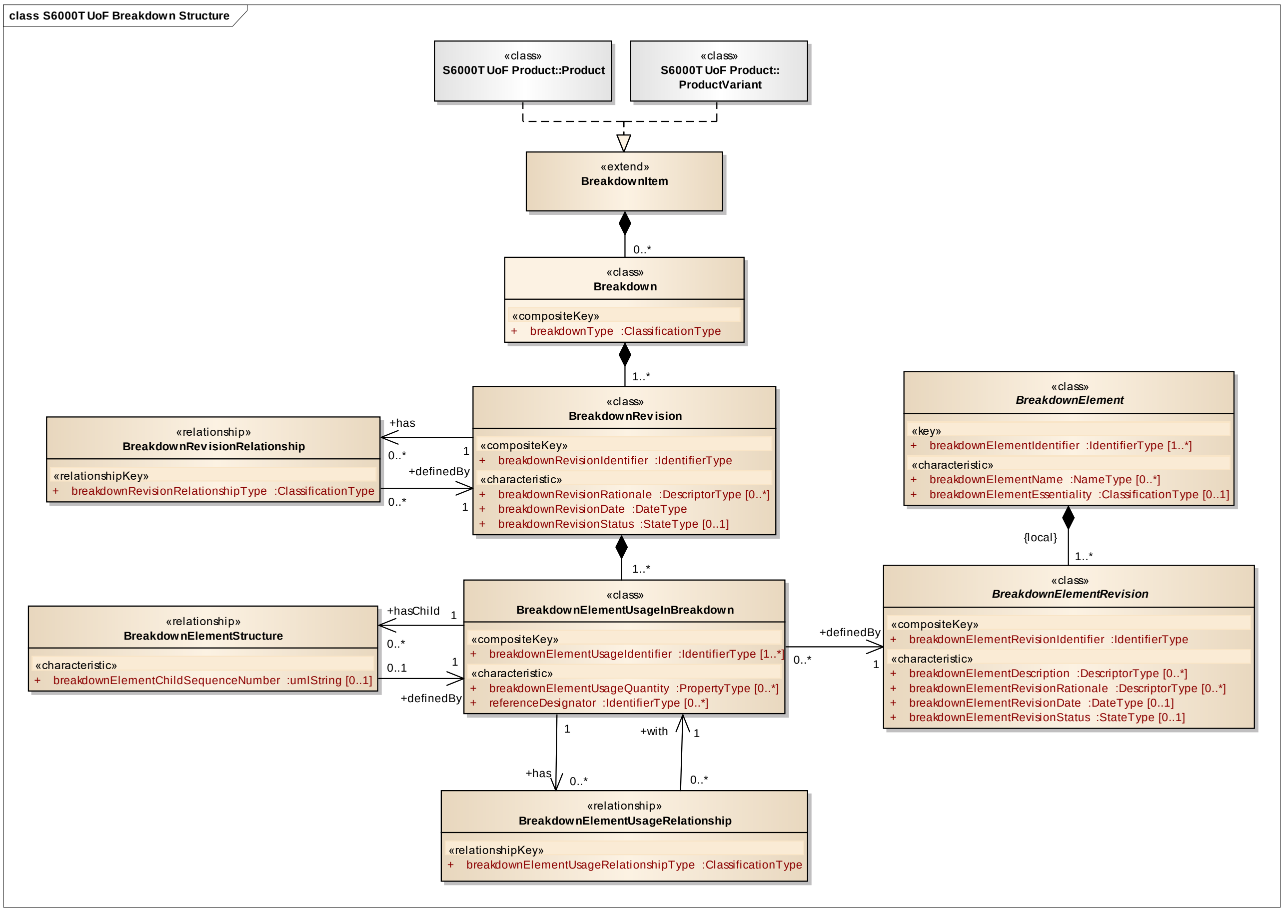
There must be one instance of `BreakdownElementUsageInBreakdown` that has no parent element. This instance of `BreakdownElementUsageInBreakdown` is often referred to as the root node.

Note

The Breakdown Structure UoF allows for a defined breakdown element and its revisions to be part of many breakdowns and/or revisions thereof. A breakdown element (revision) can be positioned differently in the respective breakdown and/or revision thereof.

Note

Each instance of BreakdownElementUsageInBreakdown and BreakdownElementStructure is uniquely defined in the context of a specific breakdown revision.



ICN-S6000T-B6865-00029-002-01

Fig 1 UoF Breakdown Structure

2.1 Breakdown

2.1.1 Class definition

Breakdown is a <<class>> that identifies a specific partitioning of a Product to form a parent-child structure of related instances of BreakdownElement.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of BreakdownRevision

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.

- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 breakdownType

breakdownType is a classification that identifies the perspective from which the Breakdown is defined.

Valid values:

- ASD (SX001G:asdSystemHardwareBreakdown)
- FAM (SX001G:familyBreakdown)
- FU (SX001G:functionalBreakdown)
- HY (SX001G:hybridBreakdown)
- PH (SX001G:physicalBreakdown)
- PR (SX001G:provisioningBreakdown)
- SY (SX001G:systemBreakdown)
- ZONE (SX001G:zonalBreakdown)

Source:

- SX001G

2.2 BreakdownElement

2.2.1 Class definition

BreakdownElement is a <<class>> defining a partition of a Product that is used in one or many instances of Breakdown.

Source:

- SX001G

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of BreakdownElementRevision

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem. Refer to Chap 7.3.46.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.2.2 Attributes

2.2.2.1 breakdownElementIdentifier (one or many)

breakdownElementIdentifier is an identifier that establishes a unique designator for a BreakdownElement and to differentiate it from other instances of BreakdownElement.

Note

Can be used to establish a hierarchical structure of the technical system.

Examples:

- the combination of Logistics Support Analysis Control Number (LCN) and Alternate Logistics Support Analysis Control Number (ALC) within GEIA-STD-0007
- the Standard Numbering System (SNS) defined by S1000D

Valid identifier class values:

- ASD (SX001G:asdSystemHardwareIdentificationCode)
- CSN (SX001G:figureItemIdentifier)
- ID (SX001G:breakdownElementIdentifier)
- LCN (SX001G:fullLogisticsSupportAnalysisControlNumber)
- SNS (SX001G:standardNumberingSystemIdentifier)

Source:

- SX001G

2.2.2.2 breakdownElementName (zero, one or many)

breakdownElementName is a name by which the BreakdownElement is known and can be easily referenced.

Source:

- SX001G

2.2.2.3 breakdownElementEssentiality (zero or one)

breakdownElementEssentiality is a classification that identifies the operational importance of the BreakdownElement at the Product level.

Note

Based on the criticality as defined during the FMECA.

Valid values:

- 1 (SX001G:criticalBreakdownElement)
- 2 (SX001G:partialCriticalBreakdownElement)
- 3 (SX001G:nonCriticalBreakdownElement)

Source:

- SX001G

2.3 BreakdownElementRevision

2.3.1 Class definition

BreakdownElementRevision is a <<class>> representing an iteration applied to a BreakdownElement.

Source:

- SX001G

2.3.1.1 Implementations

This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem. Refer to Chap 7.3.46.
- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 breakdownElementRevisionIdentifier

breakdownElementRevisionIdentifier is an identifier that establishes a unique designator for a BreakdownElementRevision and to differentiate it from other instances of BreakdownElementRevision.

Valid identifier class values:

- ID (SX001G:breakdownElementRevisionIdentifier)

Source:

- SX001G

2.3.2.2 breakdownElementDescription (zero, one or many)

breakdownElementDescription is a description that gives more information on the BreakdownElement.

Source:

- SX001G

2.3.2.3 breakdownElementRevisionRationale (zero, one or many)

breakdownElementRevisionRationale is a description that gives more information on the justification for revising the BreakdownElement.

Source:

- SX001G

2.3.2.4 breakdownElementRevisionDate (zero or one)

breakdownElementRevisionDate is a date that specifies when the BreakdownElement was revised.

Source:

- SX001G

2.3.2.5 breakdownElementRevisionStatus (zero or one)

breakdownElementRevisionStatus is a state that identifies the maturity of a BreakdownElementRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)

– R (SX001G:reviewedStatus)

Source:

– SX001G

2.4 BreakdownElementStructure

2.4.1 Class definition

BreakdownElementStructure is a <<relationship>> that establishes a hierarchical structure between two usages of BreakdownElement that belong to the same BreakdownRevision.

Source:

– SX001G

2.4.1.1 Associations

This class has the following associations:

– A directed definedBy association with one instance of BreakdownElementUsageInBreakdown

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

– DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
– ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 breakdownElementChildSequenceNumber (zero or one)

breakdownElementChildSequenceNumber is a string of characters that controls the order for the included child element.

Note

The sequence number can be used to control how child elements are presented in eg, a list.

Source:

– SX001G

2.5 BreakdownElementUsageInBreakdown

2.5.1 Class definition

BreakdownElementUsageInBreakdown is a <<class>> that represents a member of a BreakdownRevision.

Note

A BreakdownElementRevision can belong to multiple BreakdownRevisions.

Source:

– SX001G

2.5.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of BreakdownElementRevision
- A directed hasChild association with zero, one or many instances of BreakdownElementStructure
- A directed has association with zero, one or many instances of BreakdownElementUsageRelationship

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- BreakdownElementInZoneItem. Refer to Chap 7.3.46.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 breakdownElementUsageIdentifier (one or many)

breakdownElementUsageIdentifier is an identifier that establishes a unique designator for a BreakdownElementUsageInBreakdown and to differentiate it from other instances of BreakdownElementUsageInBreakdown.

Valid identifier class values:

- ID (SX001G:breakdownElementUsageIdentifier)

Source:

- SX001G

2.5.2.2 breakdownElementUsageQuantity (zero, one or many)

reakdownElementUsageQuantity is a property that specifies the amount of the BreakdownElement used in its parent BreakdownElement.

Note

If no value is given, it must be interpreted as value "1" with a unit of "each". For as required amounts, the text property is used with "As Required" or other text as appropriate.

Valid unit value:

- countUnit

Source:

- SX001G

2.5.2.3 referenceDesignator (zero, one or many)

referenceDesignator is an identifier that establishes a unique designator for a location within the overall Product, and to differentiate it from other locations.

Example:

- reference designators serve as a cross reference between parts contained in wiring diagrams, hydraulic systems etc. and the Illustrated Parts Data (IPD).

Valid identifier class values:

- RFD (SX001G:referenceDesignator)

Source:

- SX001G

2.6 BreakdownElementUsageRelationship

2.6.1 Class definition

BreakdownElementUsageRelationship is a <<relationship>> where one usage of a BreakdownElement relates to the usage of another BreakdownElement.

Note

Both related instances of BreakdownElementUsageInBreakdown must reside within the same BreakdownRevision.

Example:

- version C of a radio is restricted to the use of software version B in breakdown revision 2

Source:

- SX001G

2.6.1.1 Associations

This class has the following associations:

- A directed with association with one instance of BreakdownElementUsageInBreakdown

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 breakdownElementUsageRelationshipType

breakdownElementUsageRelationshipType is a classification that identifies the meaning of the established relationship.

Valid values:

- AND (SX001G:mutualBreakdownElementInclusion)
- XOR (SX001G:mutualBreakdownElementExclusion)

Source:

- SX001G

2.7 BreakdownItem

2.7.1 Interface definition

BreakdownItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.7.1.1 Associations

The BreakdownItem <<extend>> interface has the following associations:

- An aggregate association with zero, one or many instances of Breakdown

2.7.1.2 Class members

Classes that implement the BreakdownItem <<extend>> interface are:

- Product. Refer to Chap 7.3.23.
- ProductVariant. Refer to Chap 7.3.23.

2.8 BreakdownRevision

BreakdownRevision is a <<class>> representing an iteration applied to a Breakdown.

Note

BreakdownRevision is used to document design iterations and not breakdown variants.

Source:

- SX001G

2.8.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of BreakdownElementUsageInBreakdown
- A directed has association with zero, one or many instances of BreakdownRevisionRelationship

2.8.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.8.2 Attributes

2.8.2.1 breakdownRevisionIdentifier

breakdownRevisionIdentifier is an identifier that establishes a unique designator for a BreakdownRevision and to differentiate it from other instances of BreakdownRevision.

Valid identifier class values:

- ID (SX001G:breakdownRevisionIdentifier)

Source:

- SX001G

2.8.2.2 breakdownRevisionRationale (zero, one or many)
breakdownRevisionRationale is a description that gives more information on the justification for revising the Breakdown.

Source:

- SX001G

2.8.2.3 breakdownRevisionDate
breakdownRevisionDate is a date that specifies when the Breakdown was revised.

Source:

- SX001G

2.8.2.4 breakdownRevisionStatus (zero or one)
breakdownRevisionStatus is a state that identifies the maturity of a BreakdownRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

Source:

- SX001G

2.9 BreakdownRevisionRelationship

2.9.1 Class definition

BreakdownRevisionRelationship is a <<relationship>> where one BreakdownRevision relates to another BreakdownRevision.

Source:

- SX001G

2.9.1.1 Associations
This class has the following association:

A directed definedBy association with one instance of BreakdownRevision

2.9.1.2 Implementations
This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.

- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.10 Attributes

2.10.1 breakdownRevisionRelationshipType

`breakdownRevisionRelationshipType` is a classification that identifies the meaning of the established relationship.

Valid values:

- `BO` SX001G:basedOnBreakdownRevision
- `M` SX001G:matchesBreakdownRevision

Source:

- SX001G

Chapter 7.3.4

Units of functionality - UoF Change Information

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.13	Units of functionality - UoF Hardware Element
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.29	Units of functionality - UoF Software Element
Chap 7.3.32	Units of functionality - UoF Task

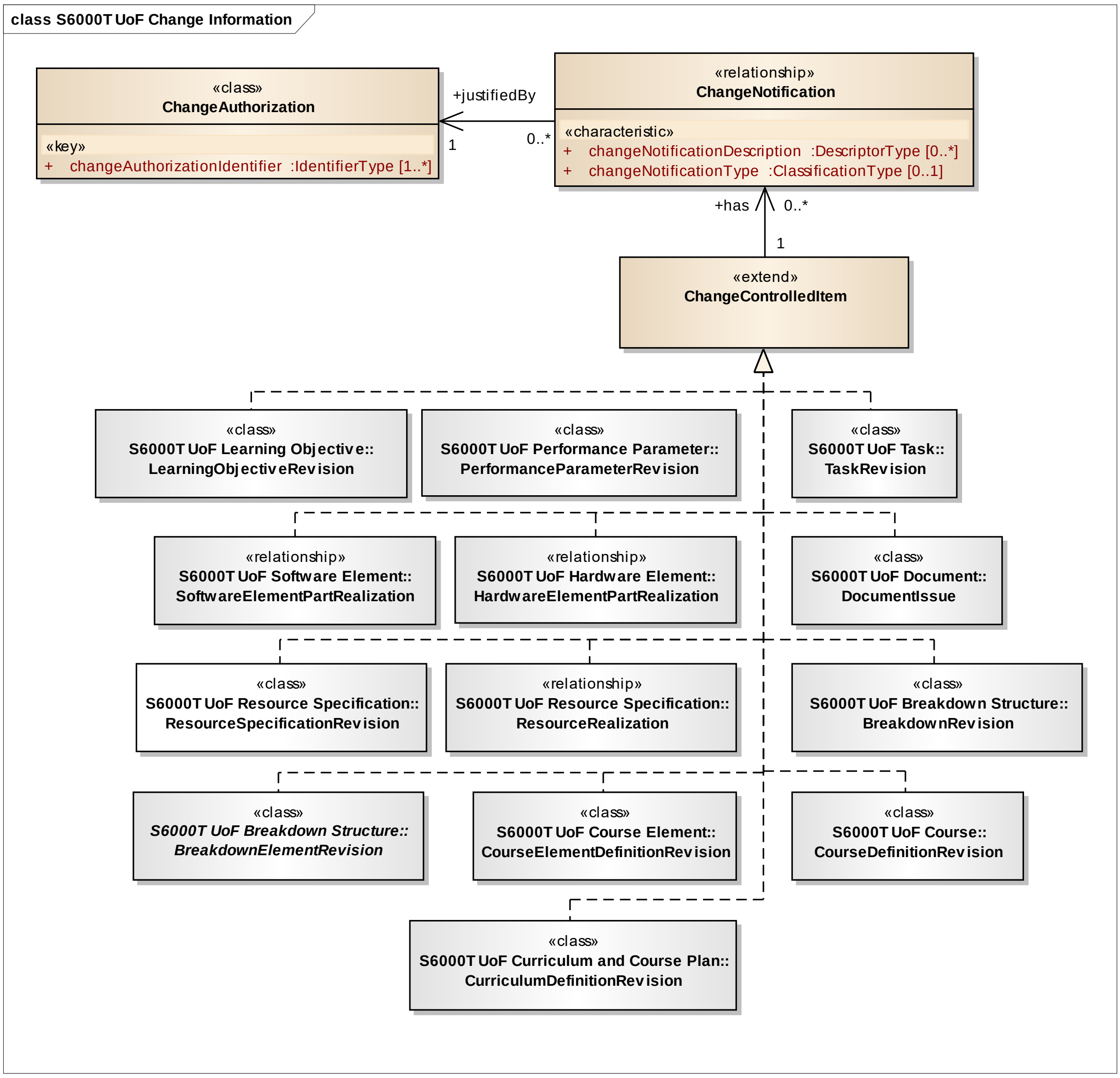
Chap No./Document No.	Title
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Change Information UoF provides the capability to identify that an item has been affected by a change authorization.

2 UoF Change Information

A key feature of the UoF Change Information data model, (refer to Fig 1), is that it is the instances of the respective class that implement the `ChangeControlledItem` `<<extend>>` interface that refers to a change authorization which justifies a creation or update of the instance and its associated information, not the other way around.



ICN-S6000T-B6865-00057-002-01

Fig 1 UoF Change Information

2.1 ChangeAuthorization

2.1.1 Class definition

ChangeAuthorization is a <<class>> that is the record of the permission to modify Product design, its procedures and/or associated Product support data.

Source:

- SX001G

2.1.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.

- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 changeAuthorizationIdentifier (one or many)

changeAuthorizationIdentifier is an identifier that establishes a unique designator for a ChangeAuthorization and to differentiate it from other instances of ChangeAuthorization.

Valid identifier class values:

- AMN (SX001G:changeAmendmentNumber)
- CAN (SX001G:changeAuthorizationNumber)
- ID (SX001G:changeAuthorizationIdentifier)

Source:

- SX001G

2.2 ChangeControlledItem

2.2.1 Interface definition

ChangeControlledItem is an <<extend>> interface that provides its associated data model to those classes that can be affected by a ChangeAuthorization.

Source:

- SX001G

2.2.1.1 Associations

The ChangeControlledItem <<extend>> interface has the following associations:

- A directed has association with zero, one or many instances of ChangeNotification

2.2.1.2 Class members

Classes that implement the ChangeControlledItem <<extend>> interface are:

- BreakdownElementRevision. Refer to Chap 7.3.3.
- BreakdownRevision. Refer to Chap 7.3.3.
- CourseDefinitionRevision. Refer to Chap 7.3.7.
- CourseElementDefinitionRevision. Refer to Chap 7.3.8.
- CurriculumDefinitionRevision. Refer to Chap 7.3.9.
- DocumentIssue. Refer to Chap 7.3.11.
- HardwareElementPartRealization. Refer to Chap 7.3.13.
- LearningObjectiveRevision. Refer to Chap 7.3.17.
- ResourceRealization. Refer to Chap 7.3.27.
- ResourceSpecificationRevision. Refer to Chap 7.3.27.
- SoftwareElementPartRealization. Refer to Chap 7.3.29.
- TaskRevision. Refer to Chap 7.3.32.

2.3 ChangeNotification

2.3.1 Class definition

ChangeNotification is a <<relationship>> that identifies an item changed due to the associated ChangeAuthorization.

Source:

– SX001G

2.3.1.1 Associations

This class has the following associations:

- A directed justifiedBy association with one instance of ChangeAuthorization

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 changeNotificationDescription (zero, one or many)

changeNotificationDescription is a description providing a summary of affects made to the related item due to a ChangeAuthorization.

Source:

- SX001G

2.3.2.2 changeNotificationType (zero or one)

changeNotificationType is a classification that identifies a change effect as belonging to a group of change effects sharing a particular characteristic or set of characteristics.

Valid values:

- A (SX001G:applicabilityChangeNotification)
- E (SX001G:editorialChangeNotification)
- M (SX001G:markupChangeNotification)
- T (SX001G:technicalChangeNotification)

Source:

- SX001G

Chapter 7.3.5

Units of functionality - UoF Circuit Breaker

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.13	Units of functionality - UoF Hardware Element
Chap 7.3.26	Units of functionality - UoF Remark
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

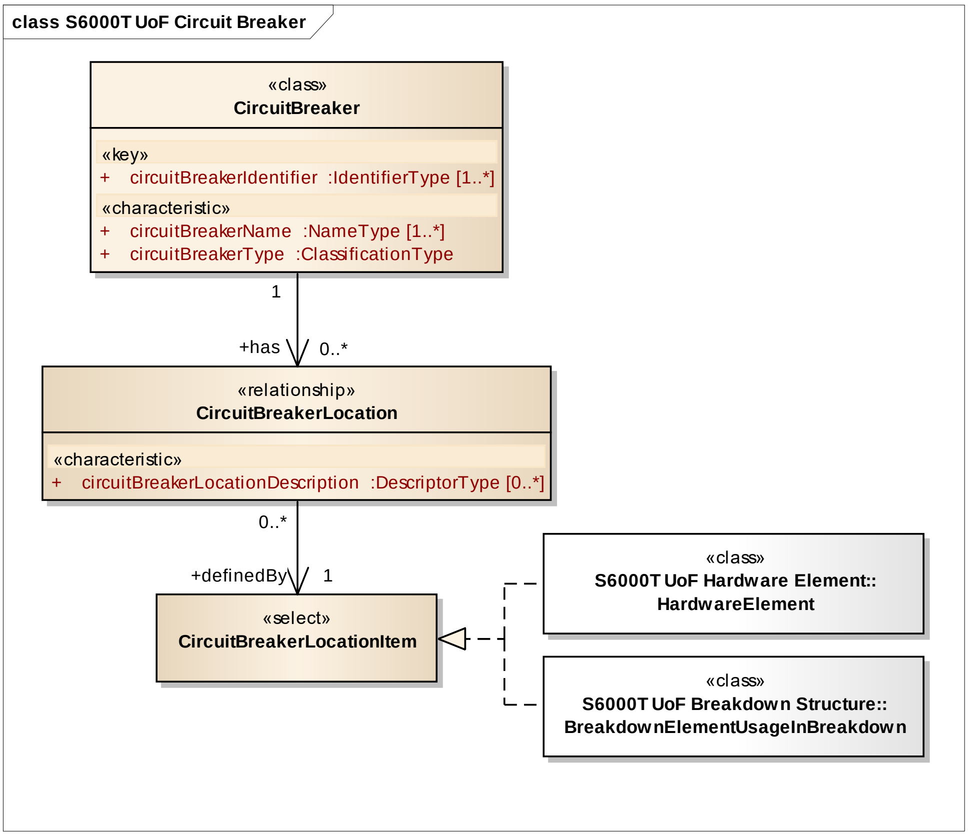
1 General

The Circuit Breaker UoF provides the capability to define circuit breakers and their locations.

2 UoF Circuit Breaker

Key features of the UoF Circuit Breaker data model, (refer to Fig 1), are:

- circuit breakers that do not qualify as individual breakdown elements in the breakdown structure for a Product can be defined and then be associated with a breakdown element representing eg, a panel
- a defined circuit breaker can have different locations eg, in different Product variants (associate `ApplicabilityStatement`s to the `CircuitBreakerLocation`)



ICN-S6000T-B6865-00095-001-01

Fig 1 UoF Circuit Breaker

2.1 CircuitBreaker

2.1.1 Class definition

`CircuitBreaker` is a `<<class>>` that represents an individual circuit breaker identified in the context of a defined Product.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- A directed has association with zero, one or many instances of CircuitBreakerLocation

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 circuitBreakerIdentifier (one or many)

circuitBreakerIdentifier is an identifier that establishes a unique designator for a CircuitBreaker and to differentiate it from other instances of CircuitBreaker.

Valid identifier class values:

- ID (SX001G:circuitBreakerSettingsIdentifier)

Source:

- SX001G

2.1.2.2 circuitBreakerName (one or many)

circuitBreakerName is a name by which the CircuitBreaker is known and can be easily referenced

Source:

- SX001G

2.1.2.3 circuitBreakerType

circuitBreakerType is a classification that defines the technical principle for the CircuitBreaker.

Valid values:

- CLIP (SX001G:dummyCircuitBreaker)
- ELMEC (SX001G:electroMechanicCircuitBreaker)
- ELTRO (SX001G:electronicCircuitBreaker)

Source:

- SX001G

2.2 CircuitBreakerLocation

2.2.1 Class definition

CircuitBreakerLocation is a <<relationship>> that identifies the item on which the CircuitBreaker is placed.

Source:

- SX001G

2.2.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CircuitBreakerLocationItem

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 circuitBreakerLocationDescription (zero, one or many)

circuitBreakerLocationDescription is a description that provides further details on where the CircuitBreaker is located on the referenced CircuitBreakerLocationItem.

Source:

- SX001G

2.3 CircuitBreakerLocationItem

2.3.1 Interface definition

CircuitBreakerLocationItem is a <<select>> interface that identifies items which can be selected in order to determine the location for a CircuitBreaker.

Source:

- SX001G

2.3.1.1 Class members

This <<select>> interface includes the following class members:

- BreakdownElementUsageInBreakdown. Refer to Chap 7.3.3.
- HardwareElement. Refer to Chap 7.3.13.

Chapter 7.3.6

Units of functionality - UoF Competency Definition

Table of contents

Page

List of tables

List of figures

References

Table 1 References

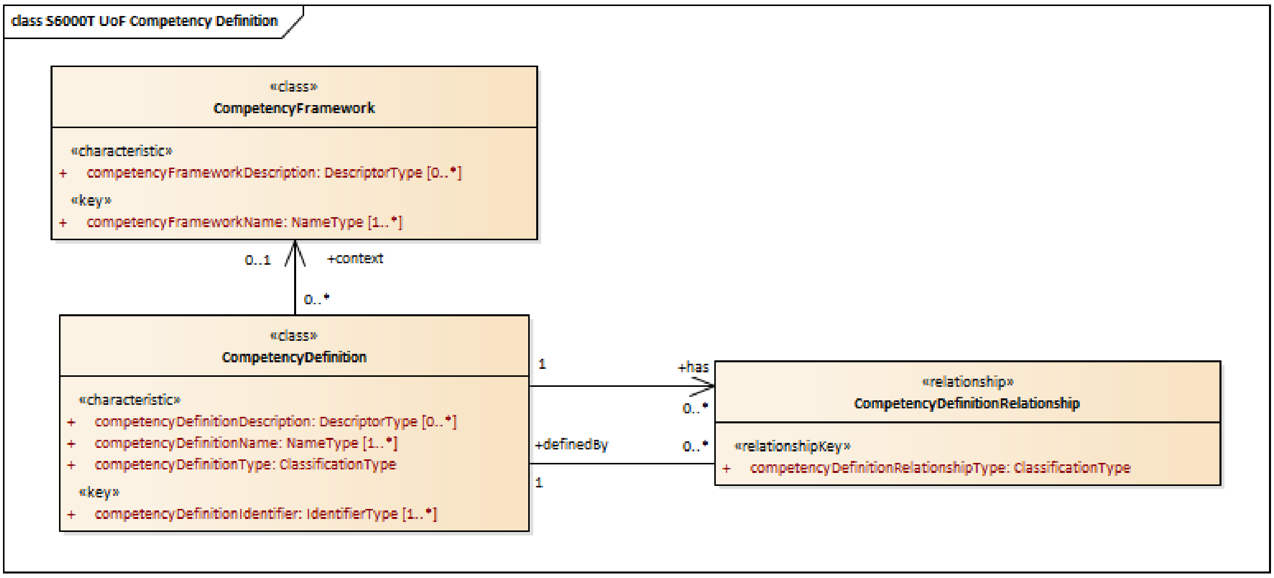
Chap No./Document No.	Title
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Competency Definition UoF provides the capability to document measurable or observable knowledge, skills, and attitudes together with the context in which they are established.

2 UoF Competency Definition

A key feature of the UoF Competency Definition data model, (refer to Fig 1), is that a competency can be defined in the context of a competency framework.



ICN-S6000T-B6865-00099-002-01

Fig 1 UoF Competency Definition

2.1 CompetencyDefinition

2.1.1 Class definition

CompetencyDefinition is a <<class>> that defines measurable or observable knowledge, skills, and attitudes necessary for successful performance by a person in a given context.

Note

Competency definition is broadly defined to include assertions of academic, professional, occupational, vocational and life goals, outcomes, and standards, however labeled.

2.1.1.1 Associations

This class has the following associations:

- A directed context association with zero or one instance of CompetencyFramework
- A directed has association with zero or one instance of CompetencyDefinitionRelationship

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from BreakdownElement). Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 competencyDefinitionIdentifier (one or many)

competencyDefinitionIdentifier is an identifier that establishes a unique designator for a CompetencyDefinition and to differentiate it from other instances of CompetencyDefinition.

Valid identifier class values:

- ID (SX001G:competencyDefinitionIdentifier)

2.1.2.2 competencyDefinitionName (one or many)

competencyDefinitionName is a name by which the CompetencyDefinition is known and can be easily referenced.

2.1.2.3 competencyDefinitionType

competencyDefinitionType is a classification that identifies further specialization of CompetencyDefinition.

Valid values:

- CR (SX001G:credentialCompetencyDefinition)
- EXP (SX001G:experienceCompetencyDefinition)
- JOB (SX001G:jobTitleCompetencyDefinition)
- QU (SX001G:qualificationCompetencyDefinition)

2.1.2.4 competencyDefinitionDescription (zero, one or many)

competencyDefinitionDescription is a description that provides information on the knowledge, skills and attitudes that a person is expected to have by having acquired a certain competency.

2.2 CompetencyFramework

2.2.1 Class definition

CompetencyFramework is a <<class>> that defines a context within which one or many CompetencyDefinitions can be defined.

Examples:

- a company providing training
- an educational institution

2.2.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 competencyFrameworkName (one or many)

competencyFrameworkName is a name by which the CompetencyFramework is known and can be easily referenced.

2.2.2.2 competencyFrameworkDescription (zero, one or many)

competencyFrameworkDescription is a description that provides further information on the CompetencyFramework.

2.3 CompetencyDefinitionRelationship

2.3.1 Class definition

CompetencyDefinitionRelationship is a <<relationship>> where one CompetencyDefinition relates to another CompetencyDefinition.

2.3.1.1 Associations

This class has the following association:

- A directed definedBy association with one instance of CompetencyDefinition

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

Chapter 7.3.7

Units of functionality - UoF Course Definition

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.19	Units of functionality - UoF Mission Definition

Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.44	Units of functionality - UoF Training System Resource Requirement
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

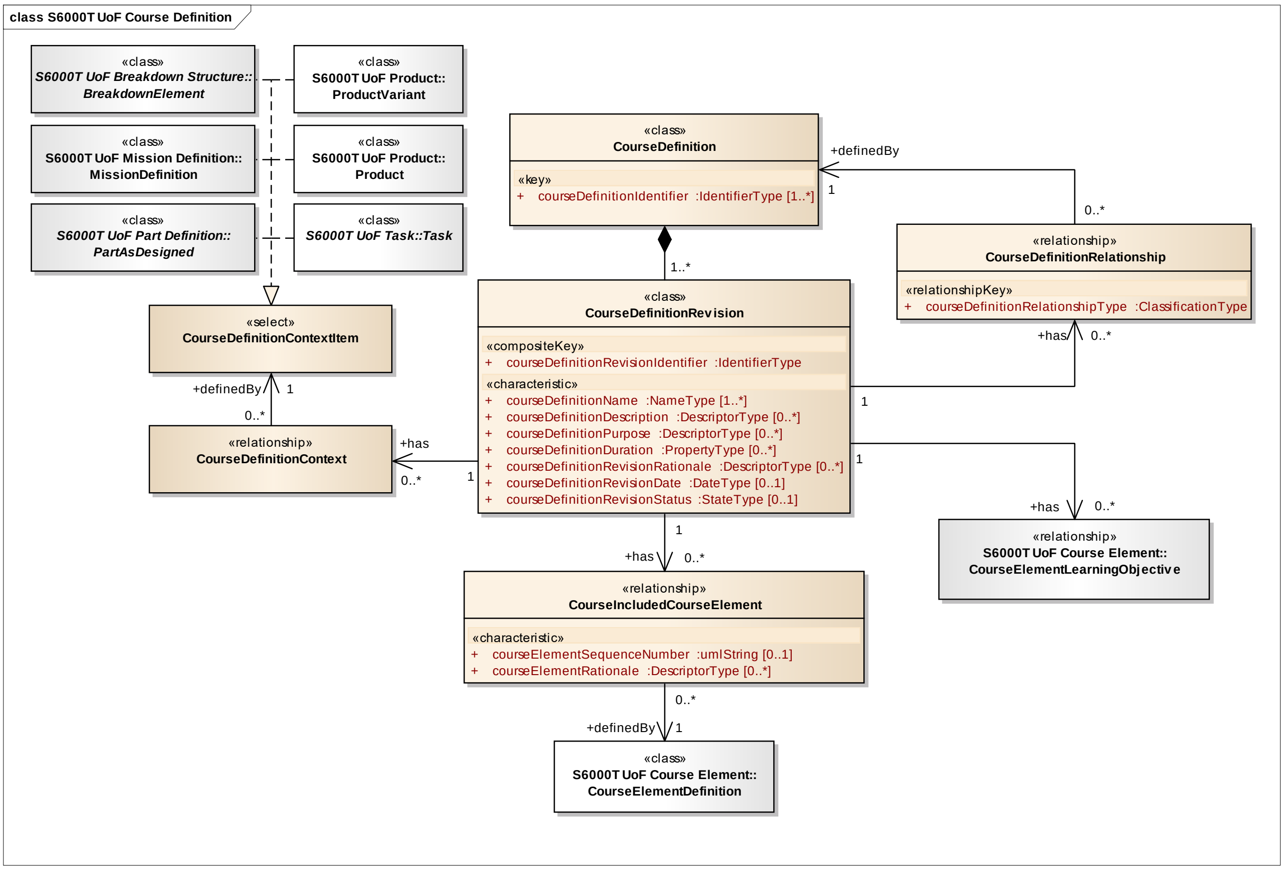
1 General

The Course Definition UoF supports sequencing of course elements into an effective instructional sequence using a course structure.

2 UoF Course Definition

Key features of the UoF Course Definition data model, (refer to Fig 1), are:

- the definition and content of a course can be iterated using `CourseDefinitionRevision`
- a course definition can be made up from both course elements as well as other course definitions
- a course can be associated with a context in which the course is defined
- a course can be associated with one or many learning objectives which are covered by, or defined for, the course definition



ICN-S6000T-B6865-00055-002-01

Fig 1 UoF Course Definition

2.1 CourseDefinition

2.1.1 Class definition

CourseDefinition is a <<class>> that defines a combination of CourseElements with the objective that a trainee is to learn knowledge, skills, and attitude to meet performance requirements for a particular aspect of a job or duty.

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of CourseDefinitionRevision

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 courseDefinitionIdentifier (one or many)

courseDefinitionIdentifier is an identifier that establishes a unique designator for a CourseDefinition and to differentiate it from other instances of CourseDefinition.

Valid identifier class values:

- ID (SX001G:courseDefinitionIdentifier)

2.2 CourseDefinitionContext

2.2.1 Class definition

CourseDefinitionContext is a <<relationship>> that identifies the item that is in focus for the CourseDefinition.

2.2.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CourseDefinitionContextItem

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3 CourseDefinitionContextItem

2.3.1 Interface definition

CourseDefinitionContextItem is a <<select>> interface that identifies items which can be selected as the context for a CourseDefinition.

2.3.1.1 Class members

This <<select>> interface includes the following class members:

- BreakdownElement Refer to Chap 7.3.3.
- MissionDefinition Refer to Chap 7.3.19.
- PartAsDesigned Refer to Chap 7.3.21.
- Product Refer to Chap 7.3.23.
- ProductVariant Refer to Chap 7.3.23.
- Task Refer to Chap 7.3.32.

2.4 CourseDefinitionRelationship

2.4.1 Class definition

CourseDefinitionRelationship is a <<relationship>> where one CourseDefinition relates to another CourseDefinition.

2.4.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CourseDefinition

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 courseDefinitionRelationshipType

courseDefinitionRelationshipType is a classification that identifies the meaning of the established relationship.

Valid values:

- A (SX001G:alternateCourseElement)
- C (SX001G:childCourseElement)
- P (SX001G:prerequisiteCourseElement)

2.5 CourseDefinitionRevision

2.5.1 Class definition

CourseDefinitionRevision is a <<class>> representing an iteration applied to a CourseDefinition.

2.5.1.1 Associations

This class has the following associations:

- A directed has association with zero, one or many instances of CourseDefinitionContext
- A directed has association with zero, one or many instances of CourseDefinitionRelationship
- A directed has association with zero, one or many instances of CourseElementLearningObjective
- A directed has association with zero, one or many instances of CourseIncludedCourseElement

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- TrainingSystemResourceRequirementItem. Refer to Chap 7.3.44.

2.5.2 Attributes

2.5.2.1 courseDefinitionRevisionIdentifier

courseDefinitionRevisionIdentifier is an identifier that establishes a unique designator for a CourseDefinitionRevision and to differentiate it from other instances of CourseDefinitionRevision.

Valid identifier class values:

- ID (SX001G:courseDefinitionRevisionIdentifier)

2.5.2.2 courseDefinitionName (one or many)

courseDefinitionName is a name by which the CourseDefinition is known and can be easily referenced.

2.5.2.3 courseDefinitionDescription (zero, one or many)

courseDefinitionDescription is a description that gives more information on the defined course.

2.5.2.4 courseDefinitionPurpose (zero, one or many)

courseDefinitionPurpose is a description that provides information about the knowledge, skills, and attitudes a trainee is intended to learn in the defined course.

2.5.2.5 courseDefinitionDuration (zero, one or many)

courseDefinitionDuration is a property that specifies the amount of time required to complete training for all course elements and learning objectives included in the CourseDefinition.

Valid unit value:

- timeUnit

2.5.2.6 courseDefinitionRevisionRationale (zero, one or many)

courseDefinitionRevisionRationale is a description that gives more information on the justification for revising the CourseDefinition.

2.5.2.7 courseDefinitionRevisionDate (zero or one)

courseDefinitionRevisionDate is a date that specifies when the CourseDefinition was revised.

2.5.2.8 courseDefinitionRevisionStatus (zero or one)

courseDefinitionRevisionStatus is a state that identifies the maturity of the CourseDefinitionRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

2.6 CourseIncludedCourseElement

2.6.1 Class definition

CourseIncludedCourseElement is a <<relationship>> that identifies a CourseElementDefinition (refer to Chap 7.3.8) that is part of the defined CourseDefinition.

2.6.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CourseElementDefinition. Refer to Chap 7.3.8.

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 courseElementSequenceNumber (zero or one)

courseElementSequenceNumber is a string value that controls the order for the CourseElementDefinition (refer to Chap 7.3.8) in relation to other CourseElementDefinitions included in the same CourseDefinition.

2.6.2.2 courseElementRationale (zero, one or many)

courseElementRationale is a description that gives more information on the justification for including the associated CourseElementDefinition (refer to Chap 7.3.8) in the CourseDefinition.

Chapter 7.3.8

Units of functionality - UoF Course Element

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Product usage phase
Chap 7.3.4	Units of functionality - UoF Hardware element
Chap 7.3.11	Units of functionality - UoF Task resource competence definition
Chap 7.3.17	Units of functionality - UoF Task train prioritization
Chap 7.3.26	Units of functionality - UoF Instructional strategy
Chap 7.3.28	Units of functionality - UoF Curriculum and course plan

Chap 7.3.41	UoF Training Media and Fidelity
Chap 7.3.44	UoF Training System Resource Requirement
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

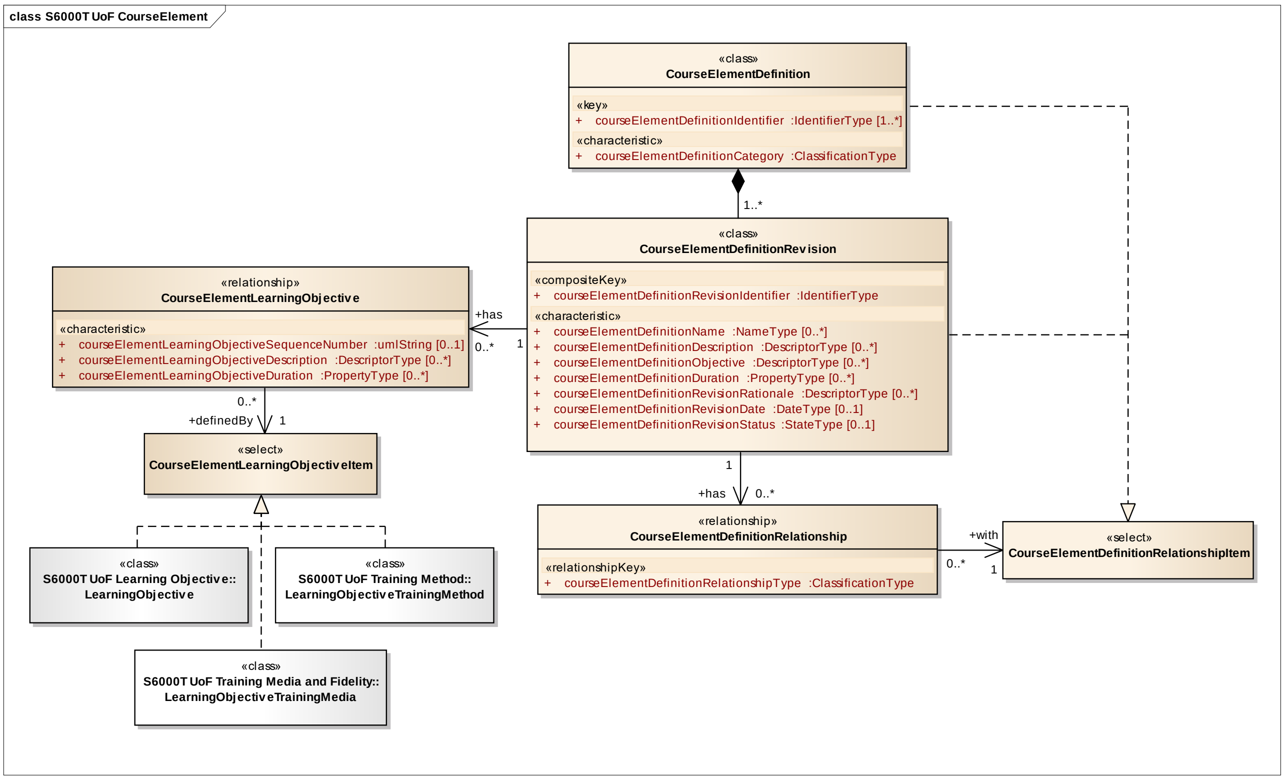
1 General

The Course Element UoF supports the sequencing of learning objectives and to document clusters and orders of the learning objectives into an effective instructional sequence using the course element structure.

2 UoF Course Element

Key features of the UoF Course Element data model, (refer to Fig 1), are:

- the definition and content of a course element can be iterated using `CourseElementDefinitionRevision`
- a course element can be made up from other course elements using `CourseElementDefinitionRelationship`
- a course element can be associated with one or many learning objectives which are covered by, or defined for, the course definition



ICN-S6000T-B6865-00103-001-01

Fig 1 UoF Course Element

2.1 CourseElementDefinition

2.1.1 Class definition

CourseElementDefinition is a <<class>> that defines a partition of a CourseDefinition where a trainee is to learn a particular knowledge, skills and/or attitude on a particular subject.

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of CourseElementDefinitionRevision

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 courseElementDefinitionIdentifier (one or many)

courseElementDefinitionIdentifier is an identifier that establishes a unique designator for a CourseElementDefinition and to differentiate it from other instances of CourseElementDefinition.

Valid identifier class values:

- ID (SX001G:courseElementDefinitionIdentifier)

2.1.2.2 courseElementDefinitionCategory

courseElementDefinitionCategory is a classification that identifies further specialization for the CourseElementDefinition and to position the CourseElementDefinition in relation to the overall CourseDefinition structure.

Valid values:

- CE (SX001G:courseElement)
- L (SX001G:courseLesson)
- M (SX001G:courseModule)
- U (SX001G:courseUnit)

2.2 CourseElementDefinitionRelationship

2.2.1 Class definition

CourseElementDefinitionRelationship is a <<relationship>> where one CourseElementDefinitionRevision relates to another CourseElementDefinition or revision thereof.

2.2.1.1 Associations

This class has the following associations:

- A directed with association with one instance of CourseElementDefinitionRelationshipItem

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 courseElementDefinitionRelationshipType

courseElementDefinitionRelationshipType is a classification that identifies the meaning of the established relationship.

Valid values:

- A (SX001G:alternateCourseElement)
- C (SX001G:childCourseElement)
- P (SX001G:prerequisiteCourseElement)

2.3 CourseElementDefinitionRelationshipItem

2.3.1 Interface definition

CourseElementDefinitionRelationshipItem is a <<select>> interface that identifies items which can be selected as the related CourseElementDefinition.

2.3.1.1 Class members

This <<select>> interface includes the following class members:

- CourseElementDefinition
- CourseElementDefinitionRevision

2.4 CourseElementDefinitionRevision

2.4.1 Class definition

CourseElementDefinitionRevision is a <<class>> representing an iteration applied to a CourseElementDefinition.

2.4.1.1 Associations

This class has the following associations:

- A directed has association with zero, one or many instances of CourseElementDefinitionRelationship
- A directed has association with zero, one or many instances of CourseElementLearningObjective

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem. Refer to Chap 7.3.4.

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- TrainingSystemResourceRequirementItem. Refer to Chap 7.3.44.

2.4.2 Attributes

2.4.2.1 courseElementDefinitionRevisionIdentifier

courseElementDefinitionRevisionIdentifier is an identifier that establishes a unique designator for a CourseElementDefinitionRevision and to differentiate it from other instances of CourseElementDefinitionRevision.

Valid identifier class values:

- ID (SX001G:courseElementDefinitionRevisionIdentifier)

2.4.2.2 courseElementDefinitionName (zero, one or many)

courseElementDefinitionName is a name by which the CourseElementDefinition is known and can be easily referenced.

2.4.2.3 courseElementDefinitionDescription (zero, one or many)

courseElementDefinitionDescription is a description that gives more information on the defined CourseElementDefinition.

2.4.2.4 courseElementDefinitionObjective (zero, one or many)

courseElementDefinitionObjective is a description that provides information about the knowledge, skills, and attitudes a trainee is intended to learn in the defined course element

2.4.2.5 courseElementDefinitionDuration (zero, one or many)

courseElementDefinitionDuration is a property that specifies the amount of time required to complete training for all learning objectives included in the CourseElementDefinition.

Valid unit value:

- timeUnit

2.4.2.6 courseElementDefinitionRevisionRationale (zero, one or many)

courseElementDefinitionRevisionRationale is a description that gives more information on the justification for revising the CourseElementDefinition.

2.4.2.7 courseElementDefinitionRevisionDate (zero or one)

courseElementDefinitionRevisionDate is a date that specifies when the CourseElementDefinition was revised.

2.4.2.8 courseElementDefinitionRevisionStatus (zero or one)

courseElementDefinitionRevisionStatus is a state that identifies the maturity of the CourseElementDefinitionRevision.

Valid state values:

- A (SX001G:approvedStatus)

- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

2.5 CourseElementLearningObjective

2.5.1 Class definition

CourseElementLearningObjective is a <<relationship>> where a CourseElementDefinition relates to a LearningObjective (refer to Chap 7.3.17) that will be fully or partially covered by the CourseElementDefinition.

2.5.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CourseElementLearningObjectiveItem

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 courseElementLearningObjectiveSequenceNumber (zero or one)

courseElementLearningObjectiveSequenceNumber is a string of characters that controls the order for the LearnigObjective (refer to Chap 7.3.17) in relation to other LearningObjectives included in the same CourseElementDefinitionRevision.

2.5.2.2 courseElementLearningObjectiveDescription (zero, one or many)

courseElementLearningObjectiveDescription is a description that gives more information about the LearningObjective (refer to Chap 7.3.17) in the context of the CourseElementDefinition.

2.5.2.3 courseElementLearningObjectiveDuration (zero, one or many)

courseElementLearningObjectiveDuration is a property that specifies the amount of time required to complete training for the associated LearningObjective (refer to Chap 7.3.17).

Valid unit value:

- timeUnit

2.6 CourseElementLearningObjectiveItem

2.6.1 Interface definition

CourseElementLearningObjectiveItem is a <<select>> interface that identifies items that represents the associated LearningObjective (refer to Chap 7.3.17) or relevant aspect thereof.

2.6.1.1 Class members

This <<select>> interface includes the following class members:

- `LearningObjective`. Refer to Chap 7.3.17.
- `LearningObjectiveTrainingMedia`. Refer to Chap 7.3.41.
- `LearningObjectiveTrainingMethod`. Refer to Chap 7.3.2.

Chapter 7.3.9

Units of functionality - UoF Curriculum and Course Plan

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.11	Units of functionality - UoF Document

Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.31	Units of functionality - UoF Target Audience
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

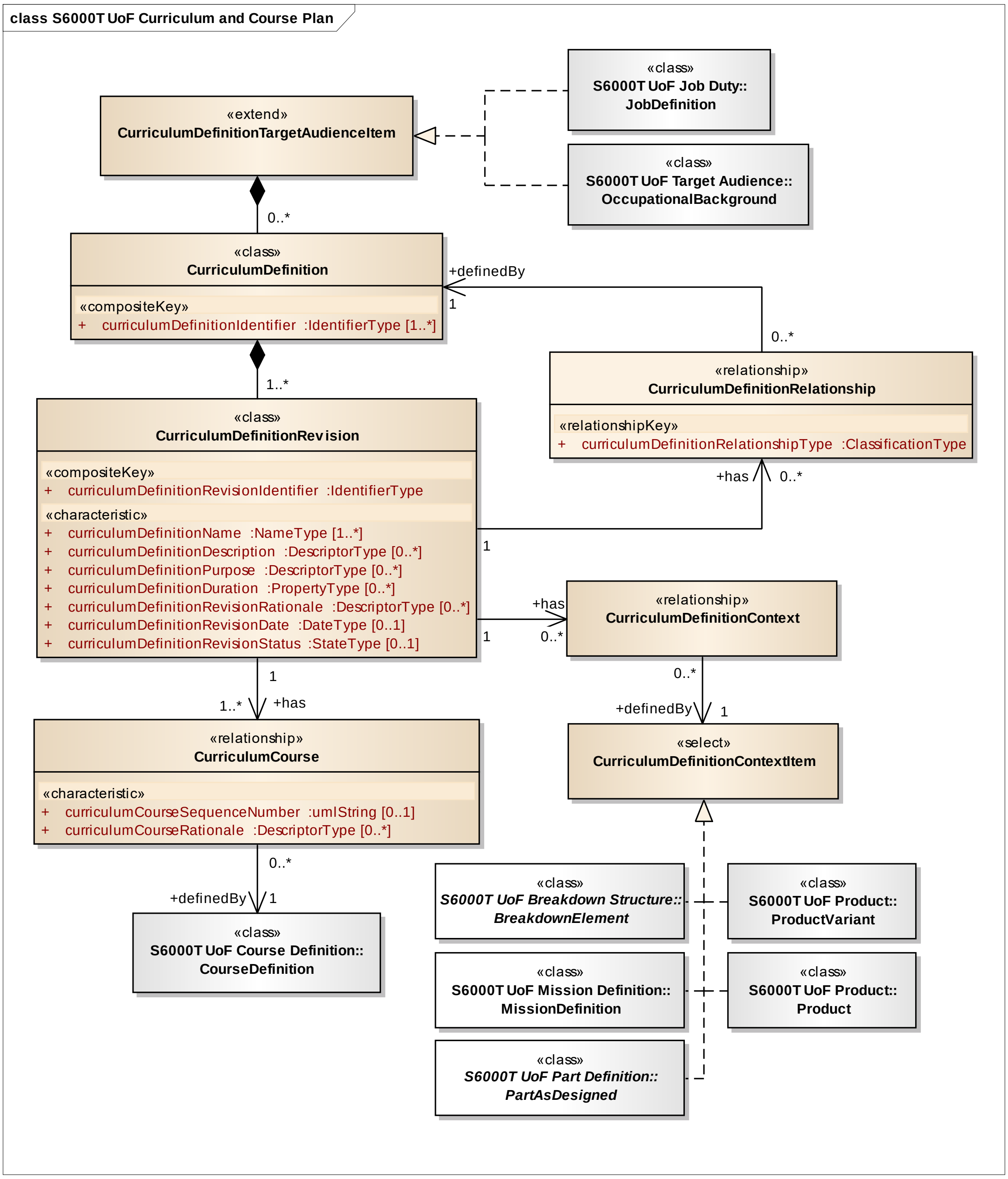
1 General

The Curriculum and Course Plan UoF supports the definition of a training path made up of courses that will give the target audience the required knowledge, skills, and attitudes.

2 UoF Curriculum and Course Plan

Key features of the UoF Curriculum and Course Plan data model, (refer to Fig 1), are:

- a curriculum can include a set of courses
- the target audience can be associated with one or many curriculums



ICN-S6000T-B6865-00054-002-01

Fig 1 UoF Curriculum and Course Plan

2.1 CurriculumCourse

2.1.1 Class definition

`CurriculumCourse` is a `<<relationship>>` that identifies a `CourseDefinition` (refer to Chap 7.3.7) that is part of the defined `CurriculumDefinition`.

2.1.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CourseDefinition. Refer to Chap 7.3.7.

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 curriculumCourseSequenceNumber (zero or one)

curriculumCourseSequenceNumber is a string value that controls the order for the CourseDefinition (refer to Chap 7.3.7) in relation to other CourseDefinitions included in the same CurriculumDefinitionRevision.

2.1.2.2 curriculumCourseRationale (zero, one or many)

curriculumCourseRationale is a description that gives more information on the justification for including the CourseDefinition (refer to Chap 7.3.7) in the CurriculumDefinition.

2.2 CurriculumDefinition

2.2.1 Class definition

CurriculumDefinition is a <class>> that defines the training path for how a particular target audience will be trained to meet the requirements for a particular job, duty, or portion thereof.

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of CurriculumDefinitionRevision

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.2.2 Attributes

2.2.2.1 curriculumDefinitionIdentifier (one or many)

curriculumDefinitionIdentifier is an identifier that establishes a unique designator for a CurriculumDefinition and to differentiate it from other instances of CurriculumDefinition.

Valid identifier class values:

– ID (SX001G:curriculumDefinitionIdentifier)

2.3 CurriculumDefinitionContext

2.3.1 Class definition

CurriculumDefinitionContext is a <<relationship>> that identifies the item that is in focus for the CurriculumDefinition.

2.3.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CurriculumDefinitionContextItem

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4 CurriculumDefinitionContextItem

2.4.1 Interface definition

CurriculumDefinitionContextItem is a <<select>> interface that identifies items which can be selected as the context for a CurriculumDefinition.

2.4.1.1 Class members

This <<select>> interface includes the following class members:

- BreakdownElement. Refer to Chap 7.3.3.
- MissionDefinition. Refer to Chap 7.3.19.
- PartAsDesigned. Refer to Chap 7.3.21.
- Product. Refer to Chap 7.3.23.
- ProductVariant. Refer to Chap 7.3.23.

2.5 CurriculumDefinitionRelationship

2.5.1 Class definition

CurriculumDefinitionRelationship is a <<relationship>> where one CurriculumDefinition relates to another CurriculumDefinition.

2.5.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CurriculumDefinition

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.

- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 curriculumDefinitionRelationshipType

curriculumDefinitionRelationshipType is a classification that identifies the meaning of the established relationship.

Example:

- variant of

Valid values:

- C (SX001G:prerequisiteCurriculum)
- S (SX001G:supportingCurriculum)
- V (SX001G:variantOfCurriculum)

2.6 CurriculumDefinitionRevision

2.6.1 Class definition

CurriculumDefinitionRevision is a <<class>> representing an iteration applied to a CurriculumDefinition.

2.6.1.1 Associations

This class has the following associations:

- A directed has association with one or many instances of CurriculumCourse
- A directed has association with zero, one or many instances of CurriculumDefinitionContext
- A directed has association with zero, one or many instances of CurriculumDefinitionRelationship

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 curriculumDefinitionRevisionIdentifier

curriculumDefinitionRevisionIdentifier is an identifier that establishes a unique designator for a CurriculumDefinitionRevision and to differentiate it from other instances of CurriculumDefinitionRevision.

Valid identifier class values:

- ID (SX001G:curriculumDefinitionRevisionIdentifier)

2.6.2.2 curriculumDefinitionName (one or many)

curriculumDefinitionName is a name by which the CurriculumDefinition is known and can be easily referenced.

2.6.2.3 curriculumDefinitionDescription (zero, one or many)
`curriculumDefinitionDescription` is a description that gives more information on the defined `CurriculumDefinition`.

2.6.2.4 curriculumDefinitionPurpose (zero, one or many)
`curriculumDefinitionPurpose` is a description of the learning outcome for a trainee completing the defined `CurriculumDefinition`.

2.6.2.5 curriculumDefinitionDuration (zero, one or many)
`curriculumDefinitionDuration` is a property that specifies the amount of time required to complete training for all courses included in the `CurriculumDefinition`.

Valid unit value:

- timeUnit

2.6.2.6 curriculumDefinitionRevisionRationale (zero, one or many)
`curriculumDefinitionRevisionRationale` is a description that gives more information on the justification for revising the `CurriculumDefinition`.

2.6.2.7 curriculumDefinitionRevisionDate (zero or one)
`curriculumDefinitionRevisionDate` is a date that specifies when the `CurriculumDefinition` was revised.

2.6.2.8 curriculumDefinitionRevisionStatus (zero or one)
`curriculumDefinitionRevisionStatus` is a state that identifies the maturity of the `CurriculumDefinitionRevision`.

Valid state values:

- `A` (SX001G:approvedStatus)
- `D` (SX001G:draftStatus)
- `IW` (SX001G:inWorkStatus)
- `C` (SX001G:cancelledStatus)
- `R` (SX001G:reviewedStatus)

2.7 CurriculumDefinitionTargetAudienceItem

2.7.1 Interface definition

`CurriculumDefinitionTargetAudienceItem` is an `<<extend>>` interface that provides its associated data model to those classes that implement it.

2.7.1.1 Class members
This `<<extend>>` interface includes the following class members:

- `JobDefinition`. Refer to Chap 7.3.14.
- `OccupationalBackground`. Refer to Chap 7.3.31.

Chapter 7.3.10

Units of functionality - UoF Digital File

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.5	Units of functionality - UoF Circuit Breaker
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.19	Units of functionality - UoF Mission Definition

Chap No./Document No.	Title
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

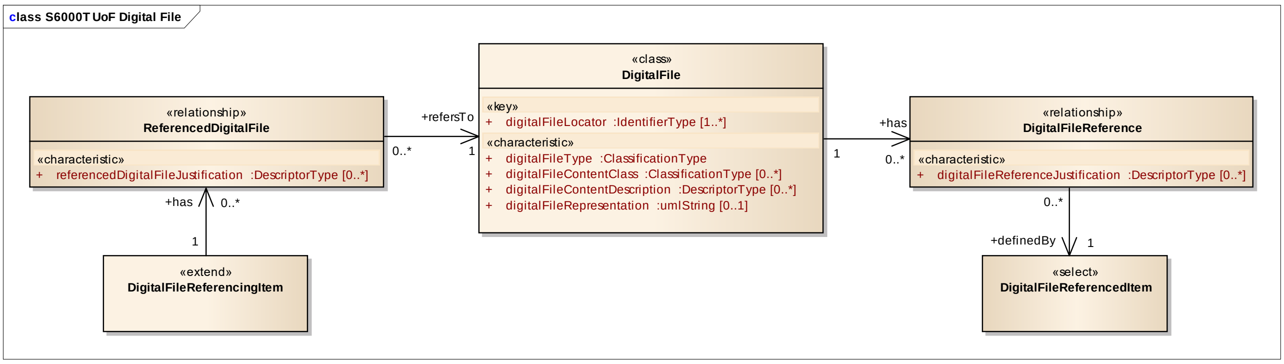
1 General

The Digital File UoF provides the capability to both reference a digital file from the exchanged data as well as to exchange the digital file itself.

2 UoF Digital File

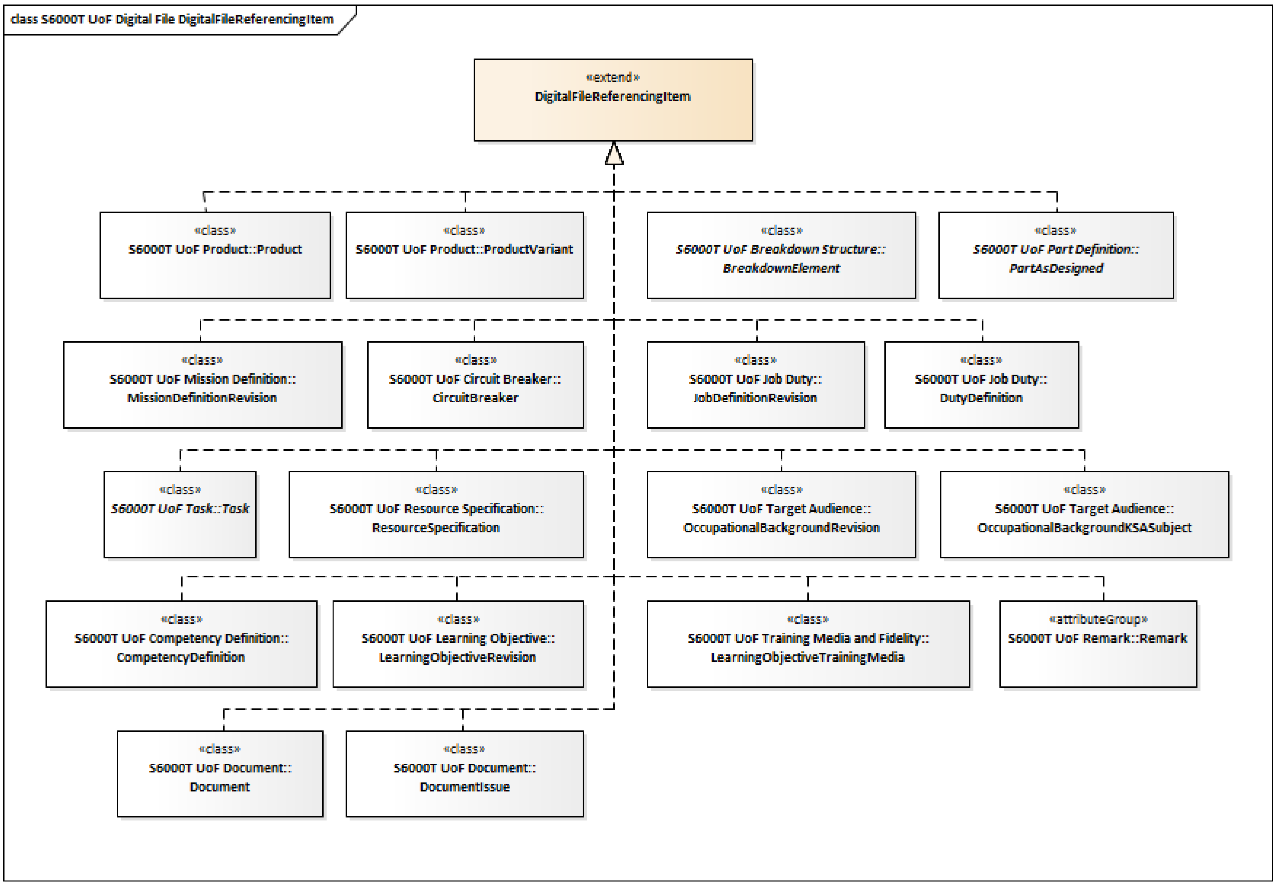
Key features of the UoF Digital File data model (refer to Fig 1, Fig 2 and Fig 3) are:

- any instance of the classes listed in Fig 2 can refer to a `DigitalFile`
- a `DigitalFile` can refer to instances of the classes listed in Fig 3



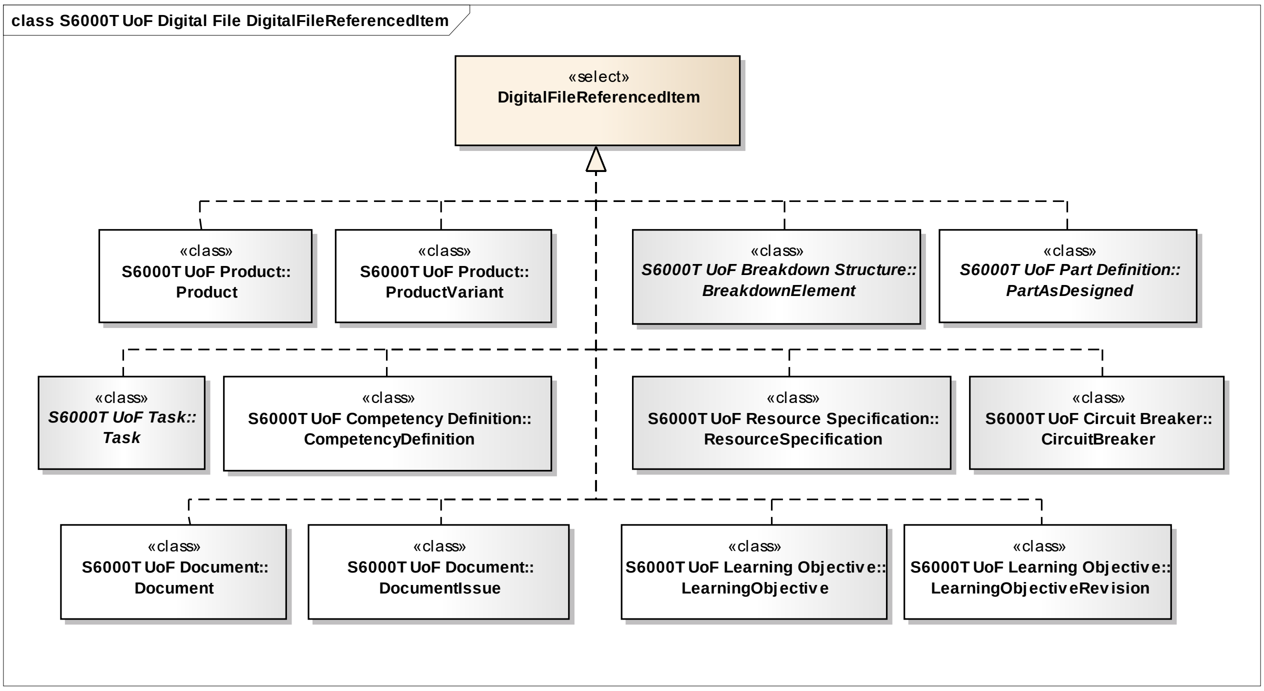
ICN-S6000T-B6865-00059-001-01

Fig 1 UoF Digital File



ICN-S6000T-B6865-00060-003-01

Fig 2 UoF Digital File - Digital File Referencing Item



ICN-S6000T-B6865-00061-002-01

Fig 3 UoF Digital File - Digital File Referenced Item

2.1 DigitalFile

2.1.1 Class definition

DigitalFile is a <<class>> that provides the identification of data stored on an electronic device that can be interpreted by a computer.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- A directed has association with zero, one or many instances of DigitalFileReference

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 digitalFileLocator (one or many)

digitalFileLocator is an identifier that establishes a unique designator for a DigitalFile used to locate and identify a DigitalFile and to differentiate it from other instances of DigitalFile.

Valid identifier class values:

- ID (SX001G:digitalFileLocator)

Source:

- SX001G

2.1.2.2 digitalFileType

digitalFileType is a classification that specifies the format of the data within the DigitalFile.

Note

Typically, the file name extension in Microsoft Windows.

Valid values:

- ASF (SX001G:advanceSystemsFormatFileType)
- AVI (SX001G:audioVideoInterleavedFileType)
- BIN (SX001G:binaryFileType)
- CGM (SX001G:computerGraphicsMetafileFileType)
- DAT (SX001G:dataFormatFileType)
- DOC (SX001G:microsoftWordFormatFileType)
- DOCX (SX001G:microsoftWordFormatFileType)

- EDF (SX001G:europeanDataFormatSignalFileType)
- HTM (SX001G:hyperTextMarkupLanguageFileType)
- JPEG (SX001G:jointPhotographicExpertsGroupFileType)
- MOV (SX001G:quickTimeMovieFileType)
- MP3 (SX001G:mp3AudioFileType)
- MPEG (SX001G:motionPictureExpertsGroupMovieFileType)
- ODS (SX001G:openDocumentSpreadsheetFileType)
- ODT (SX001G:openDocumentTextFileType)
- OTH (SX001G:otherFileType)
- PDF (SX001G:portableDocumentFormatFileType)
- PNG (SX001G:portableNetworkGraphicsFileType)
- RAW (SX001G:rawSampleAudioFileType)
- TIFF (SX001G:taggedImageFileFormatFileType)
- TXT (SX001G:textFileType)
- UNK (SX001G:unknownFileType)
- WAV (SX001G:waveformAudioFileType)
- XLS (SX001G:microsoftExcelFormatFileType)
- XLSX (SX001G:microsoftExcelFormatFileType)
- XSD (SX001G:xmlSchemaDefinitionFileType)

Source:

- SX001G

2.1.2.3 digitalFileContentClass (zero, one or many)

digitalFileContentClass is a classification that determines the meaning of the information within the DigitalFile.

Valid values:

- 3D (SX001G:3dModelFileContent)
- AUD (SX001G:audioFileContent)
- BIT (SX001G:builtInTestFileContent)
- CSW (SX001G:coursewareFileContent)
- DRW (SX001G:drawingFileContent)
- HUM (SX001G:healthAndUsageDataFileContent)
- INS (SX001G:instructionsFileContent)
- INV (SX001G:invoiceFileContent)
- MAN (SX001G:manualFileContent)
- MOV (SX001G:movieFileContent)
- OTH (SX001G:otherFileContent)
- PHO (SX001G:photographFileContent)
- PRB (SX001G:problemReportFileContent)
- PUB (SX001G:publicationFileContent)
- REP (SX001G:reportFileContent)
- SCH (SX001G:schematicsFileContent)
- SOU (SX001G:soundFileContent)
- TST (SX001G:testResultsFileContent)
- VID (SX001G:videoFileContent)
- WIR (SX001G:wiringFileContent)

Source:

- SX001G

2.1.2.4 digitalFileContentDescription (zero, one or many)
`digitalFileContentDescription` is a phrase that gives more details about the information contained in the `DigitalFile`

Source:

- SX001G

2.1.2.5 digitalFileRepresentation (zero or one)
`digitalFileRepresentation` is a string of characters representing the content of the `DigitalFile`.

Example:

- a uuencoded ASCII umlString representing a binary source file.

Source:

- SX001G

2.2 DigitalFileReference

2.2.1 Class definition

`DigitalFileReference` is a `<<relationship>>` that allows a `DigitalFile` to reference a `DigitalFileReferencedItem`.

Source:

- SX001G

2.2.1.1 Associations
This class has the following associations:

- A directed `definedBy` association with zero, one or many instances of `DigitalFileReferencedItem`

2.2.1.2 Implementations
This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 digitalFileReferenceJustification (zero, one or many)
`digitalFileReferenceJustification` is a phrase that provides more on information on the reason why the `DigitalFileReferencedItem` is referenced.

Example:

- crack discovered on `BreakdownElement` '`ABC-123`'

Source:

- SX001G

2.3 DigitalFileReferencedItem

2.3.1 Interface definition

DigitalFileReferencingItem is a <<select>> interface that identifies an item which in some way is associated with the content of DigitalFile.

Source:

- SX001G

2.3.1.1 Class members

This <<select>> interface includes the following class members:

- BreakdownElement. Refer to Chap 7.3.3.
- CircuitBreaker. Refer to Chap 7.3.5.
- CompetencyDefinition. Refer to Chap 7.3.6.
- Document. Refer to Chap 7.3.11.
- DocumentIssue. Refer to Chap 7.3.11.
- LearningObjective. Refer to Chap 7.3.17.
- LearningObjectiveRevision. Refer to Chap 7.3.17.
- PartAsDesigned. Refer to Chap 7.3.21.
- Product. Refer to Chap 7.3.23.
- ProductVariant. Refer to Chap 7.3.23.
- ResourceSpecification. Refer to Chap 7.3.27.
- Task. Refer to Chap 7.3.32.

2.4 DigitalFileReferencingItem

2.4.1 Interface definition

DigitalFileReferencingItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.4.1.1 Associations

This class has the following associations:

- A directed has association with zero, one or many instances of ReferencedDigitalFile

2.4.1.2 Class members

This <<extend>> interface includes the following class members:

- CircuitBreaker. Refer to Chap 7.3.5.
- CompetencyDefinition. Refer to Chap 7.3.6.
- Document. Refer to Chap 7.3.11.
- DocumentIssue. Refer to Chap 7.3.11.
- DutyDefinition. Refer to Chap 7.3.14.
- JobDefinitionRevision. Refer to Chap 7.3.14.
- LearningObjectiveRevision. Refer to Chap 7.3.17.

- LearningObjectiveTrainingMedia. Refer to Chap 7.3.41.
- MissionDefinitionRevision. Refer to Chap 7.3.19.
- OccupationalBackgroundKSASubject. Refer to Chap 7.3.31.
- OccupationalBackgroundRevision. Refer to Chap 7.3.31.
- PartAsDesigned. Refer to Chap 7.3.21.
- Product. Refer to Chap 7.3.23.
- ProductVariant. Refer to Chap 7.3.23.
- Remark. Refer to Chap 7.3.26.
- ResourceSpecification. Refer to Chap 7.3.27.
- Task. Refer to Chap 7.3.32.

2.5 ReferencedDigitalFile

2.5.1 Class definition

ReferencedDigitalFile is a <<relationship> that allows an item to refer to a DigitalFile.

Source:

- SX001G

2.5.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of DigitalFile

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 referencedDigitalFileJustification (zero, one or many)

referencedDigitalFileJustification is a phrase that provides more information on the reason why the DigitalFile is referenced.

Example:

- a video showing the task execution.

Source:

- SX001G

Chapter 7.3.11

Units of functionality - UoF Document

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement

Chap No./Document No.	Title
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.5	Units of functionality - UoF Circuit Breaker
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.12	Units of functionality - UoF Environment Definition
Chap 7.3.13	Units of functionality - UoF Hardware Element
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.15	Units of functionality - UoF Learning Assessment
Chap 7.3.16	Units of functionality - UoF Learning Gap
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.18	Units of functionality - UoF Message
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.20	Units of functionality - UoF Organization
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.22	Units of functionality - UoF Performance Parameter
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.24	Units of functionality - UoF Product Usage Context
Chap 7.3.25	Units of functionality - UoF Product Usage Phase
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.29	Units of functionality - UoF Software Element
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
Chap 7.3.35	Units of functionality - UoF Task Resource
Chap 7.3.36	Units of functionality - UoF Task Target

Chap No./Document No.	Title
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization
Chap 7.3.39	Units of functionality - UoF Training Entry Requirement
Chap 7.3.40	Units of functionality - UoF Training Gap
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity
Chap 7.4.42	Units of functionality - UoF Training Media Resource
Chap 7.4.43	Units of functionality - UoF Training Method
Chap 7.4.44	Units of functionality - UoF Training System Resource Requirement
Chap 7.4.45	Units of functionality - UoF Warning Caution Train Prioritization
Chap 7.4.46	Units of functionality - UoF Zone Element
S1000D	International specification for technical publications using a common source database
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

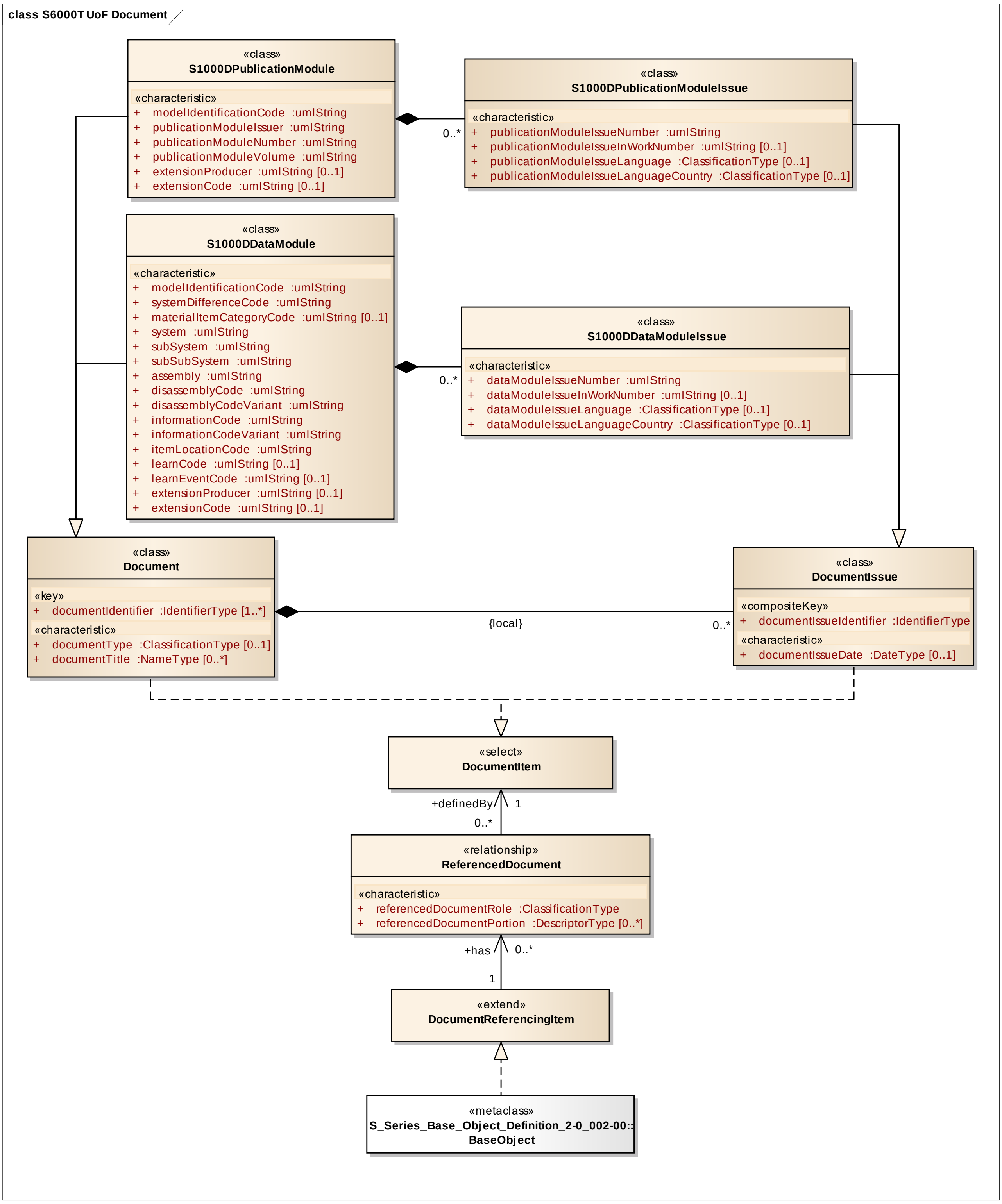
1 General

The Document UoF provides the capability to identify a physical document or a digital file and their associated metadata.

2 UoF Document

Key features of the UoF Document data model, (refer to Fig 1), are:

- any class in the data model can make a reference to a document, a specific document issue or portion thereof
- a document being referenced can either be defined to be a document in general or be a document that is published in the format of S1000D data modules or publication modules



ICN-S6000T-B6865-00058-001-01

Fig 1 UoF Document

2.1 Document

2.1.1 Class definition

Document is a <<class>> that represents a compiled set of information that serves a purpose.

Examples:

- drawing
- manual
- report

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of DocumentIssue

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.28.
- DocumentReferencingItem (inherited from BaseObject)
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 documentIdentifier (one or many)

documentIdentifier is an identifier that establishes a unique designator for a Document and to differentiate it from other instances of Document.

Valid identifier class values:

- ID (SX001G:documentIdentifier)

Source:

- SX001G

2.1.2.2 documentType (zero or one)

documentType is a classification that identifies the category of the Document.

Valid values:

- DRW (SX001G:drawingDocument)
- PCAT (SX001G:partsCatalogueDocument)
- SPEC (SX001G:specificationDocument)
- STD (SX001G:standardsDocument)
- TMAN (SX001G:technicalManual)
- TR (SX001G:technicalReport)

Source:

- SX001G

2.1.2.3 documentTitle (zero, one or many)

documentTitle is a name by which the Document is known and can be easily referenced.

Source:

- SX001G

2.2 DocumentIssue

2.2.1 Class definition

DocumentIssue is a <<class>> that represents a specific release of a Document

Source:

- SX001G

2.2.1.1 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem. Refer to Chap 7.3.4.
- DigitalFileReferencingItem. Refer to Chap 7.3.28.
- DocumentReferencingItem (inherited from BaseObject)
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 documentIssueIdentifier

documentIssueIdentifier is an identifier that establishes a unique designator for a DocumentIssue and to differentiate it from other instances of DocumentIssue.

Valid identifier class values:

- ID (SX001G:documentIssueIdentifier)

Source:

- SX001G

2.2.2.2 documentIssueDate (zero or one)

documentIssueDate is a date that defines when a specific issue of a Document was released.

Source:

- SX001G

2.3 DocumentItem

2.3.1 Interface definition

DocumentItem is a <<select>> interface that identifies items which can be selected as Document.

Source:

– SX001G

2.3.1.1 Class members

This <<select>> interface includes the following class members:

- Document
- DocumentIssue

2.4 DocumentReferencingItem

2.4.1 Interface definition

DocumentReferencingItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

– SX001G

2.4.1.1 Associations

The DocumentReferencingItem <<extend>> interface has the following associations:

A directed has association with zero, one or many, instances of ReferencedDocument

– Class members

Classes that implement the DocumentReferencingItem <<extend>> interface are:

- ApplicabilityStatement. Refer to Chap 7.3.2.
- AssociatedCompetencyDefinition. Refer to Chap 7.3.31.
- AssociatedEnvironmentDefinition. Refer to Chap 7.3.12.
- Breakdown. Refer to Chap 7.3.3.
- BreakdownElement. Refer to Chap 7.3.3.
- BreakdownElementInZone. Refer to Chap 7.3.46.
- BreakdownElementRevision. Refer to Chap 7.3.3.
- BreakdownElementStructure. Refer to Chap 7.3.3.
- BreakdownElementUsageInBreakdown. Refer to Chap 7.3.3.
- BreakdownElementUsageRelationship. Refer to Chap 7.3.3.
- BreakdownRevision. Refer to Chap 7.3.3.
- BreakdownRevisionRelationship. Refer to Chap 7.3.3.
- ChangeAuthorization. Refer to Chap 7.3.4.
- ChangeNotification. Refer to Chap 7.3.4.
- CircuitBreaker. Refer to Chap 7.3.5.
- CircuitBreakerLocation. Refer to Chap 7.3.5.
- CircuitBreakerSetting. Refer to Chap 7.3.32.
- CircuitBreakerSettings. Refer to Chap 7.3.32.
- CompetencyDefinition. Refer to Chap 7.3.6.
- CompetencyFramework. Refer to Chap 7.3.6.
- ConditionInstance. Refer to Chap 7.3.2.
- ConditionType. Refer to Chap 7.3.2.
- ConditionTypeAssertMember. Refer to Chap 7.3.2.
- CourseDefinition. Refer to Chap 7.3.7.
- CourseDefinitionContext. Refer to Chap 7.3.7.

- CourseDefinitionRelationship. Refer to Chap 7.3.7.
- CourseDefinitionRevision. Refer to Chap 7.3.7.
- CourseIncludedCourseElement. Refer to Chap 7.3.7.
- CourseElementDefinition. Refer to Chap 7.3.8.
- CourseElementDefinitionRelationship. Refer to Chap 7.3.8.
- CourseElementDefinitionRevision. Refer to Chap 7.3.8.
- CourseElementLearningObjective. Refer to Chap 7.3.8.
- CurriculumCourse. Refer to Chap 7.3.9.
- CurriculumDefinition. Refer to Chap 7.3.9.
- CurriculumDefinitionContext. Refer to Chap 7.3.9.
- CurriculumDefinitionRelationship. Refer to Chap 7.3.9.
- CurriculumDefinitionRevision. Refer to Chap 7.3.9.
- DigitalFile. Refer to Chap 7.3.10.
- DigitalFileReference. Refer to Chap 7.3.10.
- Document
- DocumentIssue
- DutyDefinition. Refer to Chap 7.3.14.
- DutyDefinitionRelationship. Refer to Chap 7.3.14.
- EnvironmentDefinition. Refer to Chap 7.3.12.
- EnvironmentDefinitionCharacteristic. Refer to Chap 7.3.12.
- EnvironmentDefinitionRelationship. Refer to Chap 7.3.12.
- EnvironmentDefinitionRevision. Refer to Chap 7.3.12.
- EvaluationCriteria. Refer to Chap 7.3.2.
- HardwareElementPartRealization. Refer to Chap 7.3.13.
- JobDefinition. Refer to Chap 7.3.14.
- JobDefinitionRevision. Refer to Chap 7.3.14.
- JobTask. Refer to Chap 7.3.14.
- KnowledgeSkillsAttitudeGapDefinition. Refer to Chap 7.3.16.
- KnowledgeSkillsAttitudeGapDefinitionRevision. Refer to Chap 7.3.16.
- LearningObjective. Refer to Chap 7.3.17.
- LearningObjectiveAdressedKsaGap. Refer to Chap 7.3.17.
- LearningObjectiveAssessmentDefinition. Refer to Chap 7.3.15.
- LearningObjectiveAssessmentTestComponent. Refer to Chap 7.3.15.
- LearningObjectiveKeyLearningPoint. Refer to Chap 7.3.17.
- LearningObjectiveRelationship. Refer to Chap 7.3.17.
- LearningObjectiveRevision. Refer to Chap 7.3.17.
- LearningObjectiveTrainingMedia. Refer to Chap 7.3.41.
- LearningObjectiveTrainingMethod. Refer to Chap 7.3.43.
- LearningObjectiveTrainingMethodIteration. Refer to Chap 7.3.43.
- MaintenanceLevel. Refer to Chap 7.3.24.
- Message. Refer to Chap 7.3.18.
- MessageContext. Refer to Chap 7.3.18.
- MessageParty. Refer to Chap 7.3.18.
- MessageRelationship. Refer to Chap 7.3.18.
- MissionDefinition. Refer to Chap 7.3.19.
- MissionDefinitionFunction. Refer to Chap 7.3.19.
- MissionDefinitionFunctionProductUsagePhase. Refer to Chap 7.3.19.

- MissionDefinitionFunctionStep. Refer to Chap 7.3.19.
- MissionDefinitionFunctionTask. Refer to Chap 7.3.19.
- MissionDefinitionParty. Refer to Chap 7.3.19.
- MissionDefinitionRelationship. Refer to Chap 7.3.19.
- MissionDefinitionRevision. Refer to Chap 7.3.19.
- OccupationalBackground. Refer to Chap 7.3.31.
- OccupationalBackgroundKSASubject. Refer to Chap 7.3.31.
- OccupationalBackgroundRevision. Refer to Chap 7.3.31.
- OperatingLocationType. Refer to Chap 7.3.24.
- Organization. Refer to Chap 7.3.20.
- PartAsDesigned. Refer to Chap 7.3.21.
- PerformanceParameter. Refer to Chap 7.3.22.
- PerformanceParameterRelationship. Refer to Chap 7.3.22.
- PerformanceParameterRevision. Refer to Chap 7.3.22.
- Product. Refer to Chap 7.3.23.
- ProductUsagePhase. Refer to Chap 7.3.25.
- ProductUsagePhaseRelationship. Refer to Chap 7.3.25.
- ProductVariant. Refer to Chap 7.3.23.
- ReferencedDigitalFile. Refer to Chap 7.3.10.
- ReferencedDocument
- ResourceRealization. Refer to Chap 7.3.27.
- ResourceSpecification. Refer to Chap 7.3.27.
- ResourceSpecificationRevision. Refer to Chap 7.3.27.
- ResultingDataModule. Refer to Chap 7.3.32.
- SecurityClass. Refer to Chap 7.3.28.
- SecurityClassification. Refer to Chap 7.3.28.
- SoftwareElementPartRealization. Refer to Chap 7.3.29.
- Subtask. Refer to Chap 7.3.32.
- SubtaskAcceptanceParameter. Refer to Chap 7.3.32.
- SubtaskInZone. Refer to Chap 7.3.32.
- SubtaskJobDefinition. Refer to Chap 7.3.30.
- SubtaskSourceDocument. Refer to Chap 7.3.32.
- SubtaskTarget. Refer to Chap 7.3.32.
- SubtaskTimeline. Refer to Chap 7.3.32.
- SubtaskTrainDecision. Refer to Chap 7.3.30.
- SubtaskTrainStep. Refer to Chap 7.3.30.
- SubtaskWarningCautionNote. Refer to Chap 7.3.32.
- Task. Refer to Chap 7.3.32.
- TaskAttitudeLevelOfLearning. Refer to Chap 7.3.33.
- TaskKnowledgeLevelOfLearning. Refer to Chap 7.3.33.
- TaskKnowledgeSkillsAttitudeGap. Refer to Chap 7.3.16.
- TaskResource. Refer to Chap 7.3.35.
- TaskResourceRelationship. Refer to Chap 7.3.35.
- TaskRevision. Refer to Chap 7.3.32.
- TaskSkillLevelOfLearning. Refer to Chap 7.3.33.
- TaskTarget. Refer to Chap 7.3.36.
- TaskTrainDecision. Refer to Chap 7.3.37.
- TaskRevisionWarningCautionNote. Refer to Chap 7.3.32.

- TrainingEntryCompetencyDefinition. Refer to Chap 7.3.39.
- TrainingEntryRequirement. Refer to Chap 7.3.39.
- TrainingGap. Refer to Chap 7.3.40.
- TrainingMediaResource. Refer to Chap 7.3.42.
- TrainingSystemResourceRequirement. Refer to Chap 7.3.44.
- WarningCautionNote. Refer to Chap 7.3.32.
- WarningCautionTrainDecision. Refer to Chap 7.3.45.
- WarningCautionTrainJob. Refer to Chap 7.3.45.

2.5 ReferencedDocument

2.5.1 Class definition

ReferencedDocument is a <<relationship>> where one DocumentReferencingItem relates to a DocumentItem.

Source:

- SX001G

2.5.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of DocumentItem

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject)
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 referencedDocumentRole

referencedDocumentRole is a classification that identifies the function of the established relationship.

Examples:

- design document reference
- directive
- document reference
- drawing reference
- source
- verification

Valid values:

- DES (SX001G:designDocumentReference)
- DIR (SX001G:directiveDocumentReference)
- DRW (SX001G:drawingDocumentReference)
- REF (SX001G:generalDocumentReference)
- REQ (SX001G:requirementsDocumentReference)
- RES (SX001G:resultDocumentReference)

- SRC (SX001G:sourceDocumentReference)
- VAL (SX001G:validationDocumentReference)
- VER (SX001G:verificationDocumentReference)

Source:

- SX001G

2.5.2.2 referencedDocumentPortion (zero, one or many)
referencedDocumentPortion is a description that provides a reference to the portion of a Document which is of interest in a specific usage.

Source:

- SX001G

2.6 S1000DDataModule

2.6.1 Class definition

S1000DDataModule is a Document that is written in accordance with an S1000D schema.

Source:

- SX001G

2.6.1.1 Associations
This class has the following associations:

- An aggregate association with zero, one or many instances of S1000DDataModuleIssue

2.6.1.2 Implementations
This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem (inherited from Document). Refer to Chap 7.3.28.
- DocumentReferencingItem (inherited from BaseObject)
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from Document). Refer to Chap 7.3.28.

2.6.2 Attributes

2.6.2.1 documentIdentifier (one or many
Inherited from Document.

2.6.2.2 documentType (zero or one)
Inherited from Document.

2.6.2.3 documentTitle (zero, one or many)
Inherited from Document.

2.6.2.4 modelIdentificationCode

`modelIdentificationCode` is a string of characters that represents the model identification code attribute of the data module code.

Note

A `modelIdentificationCode` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.5 systemDifferenceCode

`systemDifferenceCode` is a string of characters that represents the system difference code attribute of the data module code.

Note

A `systemDifferenceCode` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.6 materialItemCategoryCode (zero or one)

`materialItemCategoryCode` is a string of characters that represents the material item category code attribute of the data module code.

Note

A `materialItemCategoryCode` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.7 system

`system` is a string of characters that represents the system attribute of the data module code.

Note

A `system` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.8 subsystem

`subSystem` is a string of characters that represents the subsystem attribute of the data module code.

Note

A `subSystem` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.9 subSubSystem

`subSubSystem` is a string of characters that represents the sub-subsystem attribute of the data module code.

Note

A `subSubSystem` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.10 assembly

`assembly` is a string of characters that represents the unit or assembly attribute of the data module code.

Source:

- SX001G

2.6.2.11 disassemblyCode

`disassemblyCode` is a string of characters that represents the disassembly code attribute of the data module code.

Note

A `disassemblyCode` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.12 disassemblyCodeVariant

`disassemblyCodeVariant` is a string of characters that represents the disassembly code variant attribute of the data module code.

Note

A `disassemblyCodeVariant` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.13 informationCode

`informationCode` is a string of characters that represents the information code attribute of the data module code.

Note

An `informationCode` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.14 informationCodeVariant

`informationCodeVariant` is a string of characters that represents the information code variant attribute of the data module code.

Note

An `informationCodeVariant` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.15 itemLocationCode

`itemLocationCode` is a string of characters that represents the item location code attribute of the data module code.

Note

An `itemLocationCode` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.16 learnCode (zero or one)

`learnCode` is a string of characters that represents the learn code attribute of the data module code.

Note

A `learnCode` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.17 learnEventCode (zero or one)

`learnEventCode` is a string of characters that represents the learn event code attribute of the data module code.

Note

A `learnEventCode` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.18 extensionProducer (zero or one)

`extensionProducer` is a string of characters used to identify the organization providing the customized data module.

Note

An `extensionProducer` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.6.2.19 extensionCode (zero or one)

`extensionCode` is a string of characters used to identify the organization receiving the customized data module.

Note

An extensionCode must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.7 S1000DDataModuleIssue

2.7.1 Class definition

S1000DDataModuleIssue is a DocumentIssue that identifies a specific issue of a data module produced in accordance with S1000D.

Source:

- SX001G

2.7.1.1 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem (inherited from DocumentIssue)
- DigitalFileReferencingItem (inherited from DocumentIssue)
- DocumentReferencingItem (inherited from BaseObject)
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.7.2 Attributes

2.7.2.1 documentIssueIdentifier

Inherited from DocumentIssue.

Source:

- SX001G

2.7.2.2 documentIssueDate (zero or one)

Inherited from DocumentIssue.

Source:

- SX001G

2.7.2.3 dataModuleIssueNumber

dataModuleIssueNumber is a string of characters used to identify the release number of the S1000DDataModuleIssue

Note

A dataModuleIssueNumber must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.7.2.4 dataModuleIssueInWorkNumber (zero or one)

dataModuleIssueInWorkNumber is a string of characters used for monitoring and control of intermediate drafts of S1000DDataModuleIssue.

Note

A dataModuleIssueInWorkNumber must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.7.2.5 dataModuleIssueLanguage (zero or one)

dataModuleIssueLanguage is a classification that identifies the language used to produce the content of the S1000DDataModuleIssue.

Note

A dataModuleIssueLanguage must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.7.2.6 dataModuleIssueLanguageCountry (zero or one)

dataModuleIssueLanguageCountry is a classification that identifies the country where the language, identified by dataModuleIssueLanguage, is spoken

Note

A dataModuleIssueLanguageCountry must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.8 S1000DPublicationModule

2.8.1 Class definition

S1000DPublicationModule is a Document that identifies a publication published in accordance with S1000D.

Source:

- SX001G

2.8.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of S1000DPublicationModuleIssue

2.8.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem (inherited from Document). Refer to Chap 7.3.28.
- DocumentReferencingItem (inherited from BaseObject)
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

- SecurityClassificationItem (inherited from Document). Refer to Chap 7.3.28.

2.8.2 Attributes

2.8.2.1 documentIdentifier (one or many

Inherited from Document.

2.8.2.2 documentType (zero or one)

Inherited from Document.

2.8.2.3 documentTitle (zero, one or many)

Inherited from Document.

2.8.2.4 modelIdentificationCode

modelIdentificationCode is a string of characters that represents the model identification code attribute of the data module code.

Note

A modelIdentificationCode must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.8.2.5 publicationModuleIssuer

publicationModuleIssuer is a string of characters that represents the issuing authority attribute of the publication module code.

Note

A publicationModuleIssuer must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.8.2.6 publicationModuleNumber

publicationModuleIssueNumber is a string of characters used to identify the release number of the S1000DPublicationModuleIssue.

Note

A publicationModuleIssueNumber must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.8.2.7 publicationModuleVolume

publicationModuleVolume is a string of characters that represents the volume of the publication attribute of the publication module code.

Note

A publicationModuleVolume must be created in accordance with the rules defined in S1000D.

Source:

– SX001G

2.8.2.8 extensionProducer (zero or one)
`extensionProducer` is a string of characters used to identify the organization providing the customized data module.

Note

An `extensionProducer` must be created in accordance with the rules defined in S1000D.

Source:

– SX001G

2.8.2.9 extensionCode (zero or one)
`extensionCode` is a string of characters used to identify the organization receiving the customized data module.

Note

An `extensionCode` must be created in accordance with the rules defined in S1000D.

Source:

– SX001G

2.9 S1000DPublicationModuleIssue

2.9.1 Class definition

`S1000DPublicationModuleIssue` is a `DocumentIssue` that identifies a specific issue of a publication module published in accordance with S1000D.

Source:

– SX001G

2.9.1.1 Implementations
This class implements the following `<<extend>>` interfaces:

- `ChangeControlledItem` (inherited from `DocumentIssue`)
- `DigitalFileReferencingItem` (inherited from `DocumentIssue`)
- `DocumentReferencingItem` (inherited from `BaseObject`)
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.9.2 Attributes

2.9.2.1 documentIssueIdentifier
Inherited from `DocumentIssue`.

Source:

– SX001G

2.9.2.2 documentIssueDate (zero or one)
Inherited from `DocumentIssue`.

Source:

- SX001G

2.9.2.3 publicationModuleIssueNumber

`publicationModuleIssueNumber` is a string of characters used to identify the release number of the `S1000DPublicationModuleIssue`.

Note

A `publicationModuleIssueNumber` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.9.2.4 publicationModuleIssueInWorkNumber (zero or one)

`publicationModuleIssueInWorkNumber` is a string of characters used for monitoring and control of intermediate drafts of `S1000DPublicationModuleIssue`.

Note

A `publicationModuleIssueInWorkNumber` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.9.2.5 publicationModuleIssueLanguage (zero or one)

`publicationModuleIssueLanguage` is a classification that identifies the language used to produce the content of the `S1000DPublicationModuleIssue`.

Note

A `publicationModuleIssueLanguage` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

2.9.2.6 publicationModuleIssueLanguageCountry (zero or one)

`publicationModuleIssueLanguageCountry` is a classification that identifies the country where the language, identified by `publicationModuleIssueLanguage`, is spoken

Note

A `publicationModuleIssueLanguageCountry` must be created in accordance with the rules defined in S1000D.

Source:

- SX001G

End of data module

Chapter 7.3.12

Units of functionality - UoF Environment Definition

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chapter	Data module title
Chap 7.3.32	Units of functionality - UoF Task resource competence definition
Chap 7.3.19	Units of functionality - UoF Warning caution train prioritization
Chap 7.3.26	Units of functionality - UoF Instructional strategy
Chap 7.3.11	Units of functionality - UoF Document
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

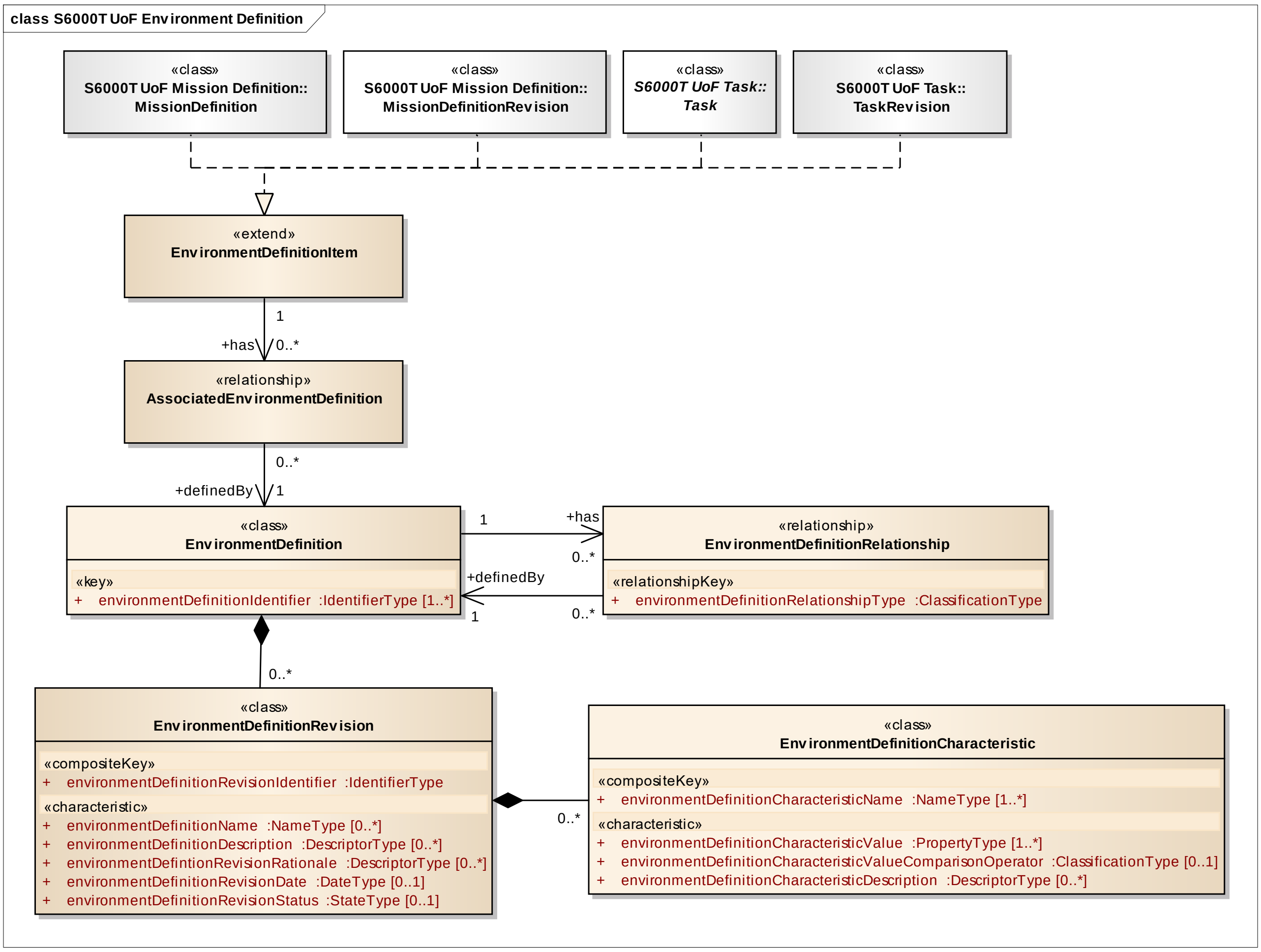
1 General

The Environment Definition UoF provides the capability to define circumstances, objects, events and/or conditions by which something can be surrounded and that influence the performance of an associated item.

2 UoF Environment Definition

Key features of the UoF Environment Definition data model, (refer to Fig 1), are:

- the definition of an `EnvironmentDefinition` and its characteristics can be iterated using `EnvironmentDefinitionRevision`
- an `EnvironmentDefinition` can have a set of measurable `EnvironmentDefinitionCharacteristic`s which can be used to determine if an existing environment qualifies as being of the `EnvironmentDefinition` or not



ICN-S6000T-B6865-00094-001-01

Fig 1 UoF Environment Definition

2.1 AssociatedEnvironmentDefinition

2.1.1 Class definition

`AssociatedEnvironmentDefinition` is a `<<relationship>>` that associates an `EnvironmentDefinitionItem` with an `EnvironmentDefinition` relevant to its existence, operation and/or support.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of EnvironmentDefinition

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2 EnvironmentDefinition

2.2.1 Class definition

EnvironmentDefinition is a <<class>> that specifies the circumstances, objects, events and/or conditions by which something can be surrounded and that influence the performance of an associated item.

Source:

- SX001G

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of EnvironmentDefinitionRevision
- A directed has association with zero, one or many instances of EnvironmentDefinitionRelationship

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 environmentDefinitionIdentifier (one or many)

environmentDefinitionIdentifier is an identifier that establishes a unique designator for an EnvironmentDefinition and to differentiate it from other instances of EnvironmentDefinition.

Valid identifier class values:

- ID (SX001G:environmentDefinitionIdentifier)

Source:

- SX001G

2.3 EnvironmentDefinitionCharacteristic

2.3.1 Class definition

EnvironmentDefinitionCharacteristic is a <<class>> that represents a measurable or observable characteristic for a circumstance, object, event or condition that is significant to the EnvironmentDefinition.

Source:

- SX001G

2.3.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 environmentDefinitionCharacteristicName (one or many)

environmentDefinitionCharacteristicName is a name by which the EnvironmentDefinitionCharacteristic is known and can be easily referenced.

Source:

- SX001G

2.3.2.2 environmentDefinitionCharacteristicValue (one or many)

environmentDefinitionCharacteristicValue is a property that represents a measurable or observable characteristic for a circumstance, object, event or condition that is typical to the EnvironmentDefinition.

Valid unit value:

- unit

Source:

- SX001G

2.3.2.3 environmentDefinitionCharacteristicValueComparisonOperator (zero or one)

environmentDefinitionCharacteristicValueComparisonOperator is a classification that identifies the comparison operator which will be used to qualify whether an actual environment is as a member of the defined EnvironmentDefinition.

Valid values:

- EQ (SX001G:equalToComparisonOperator)
- NE (SX001G:notEqualToComparisonOperator)
- LT (SX001G:lessThanComparisonOperator)
- LE (SX001G:lessThanOrEqualToComparisonOperator)

- GT (SX001G:greaterThanComparisonOperator)
- GE (SX001G:greaterThanOrEqualToComparisonOperator)
- IN (SX001G:withinRangeComparisonOperator)
- OUT (SX001G:outsideRangeComparisonOperator)

Source:

- SX001G

2.3.2.4 environmentDefinitionCharacteristicDescription (zero, one or many)
environmentDefinitionCharacteristicDescription is a description that gives more information on the EnvironmentDefinitionCharacteristic.

Source:

- SX001G

2.4 EnvironmentDefinitionItem

2.4.1 Interface definition

EnvironmentDefinitionItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.4.1.1 Associations
The EnvironmentDefinitionItem <<extend>> interface has the following associations:

- A directed has association with zero, one or many instances of AssociatedEnvironmentDefinition

2.4.1.2 Class members
This <<extend>> interface includes the following class members:

- MissionDefinition. Refer to Chap 7.3.19.
- MissionDefinitionRevision. Refer to Chap 7.3.19.
- Task. Refer to Chap 7.3.32.
- TaskRevision. Refer to Chap 7.3.32.

2.5 EnvironmentDefinitionRelationship

2.5.1 Class definition

EnvironmentDefinitionRelationship is a <<relationship>> where one EnvironmentDefinition relates to another EnvironmentDefinition.

Source:

- SX001G

2.5.1.1 Associations
This class has the following associations:

- A directed definedBy association with one instance of EnvironmentDefinition

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 environmentDefinitionRelationshipType

environmentDefinitionRelationshipType is a classification that identifies the meaning of the established relationship.

Valid values:

- SPEC (SX001G:specializationOfEnvironmentDefinition)
- SSNL (SX001G:seasonalEnvironmentDefinitionVariation)

Source:

- SX001G

2.6 EnvironmentDefinitionRevision

2.6.1 Class definition

EnvironmentDefinitionRevision is a <<class>> representing an iteration applied to an EnvironmentDefinition.

Source:

- SX001G

2.6.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of EnvironmentDefinitionCharacteristic

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 environmentDefinitionRevisionIdentifier

environmentDefinitionRevisionIdentifier is an identifier that establishes a unique designator for an EnvironmentDefinitionRevision and to differentiate it from other instances of EnvironmentDefinitionRevision.

Valid identifier class values:

- ID (SX001G:environmentDefinitionRevisionIdentifier)

Source:

- SX001G

2.6.2.2 environmentDefinitionName (zero, one or many)
`environmentDefinitionName` is a name by which the `EnvironmentDefinition` is known and can be easily referenced.

Source:

- SX001G

2.6.2.3 environmentDefinitionDescription (zero, one or many)
`environmentDefinitionDescription` is a description that gives more information on the `EnvironmentDefinition`.

Source:

- SX001G

2.6.2.4 environmentDefintionRevisionRationale (zero, one or many)
`environmentDefinitionRevisionRationale` is a description that provides a justification for revising the `EnvironmentDefinition`.

Source:

- SX001G

2.6.2.5 environmentDefinitionRevisionDate (zero or one)
`environmentDefinitionRevisionDate` is a date that specifies when the `EnvironmentDefinition` was revised.

Source:

- SX001G

2.6.2.6 environmentDefinitionRevisionStatus (zero or one)
`environmentDefinitionRevisionStatus` is a state that identifies the maturity of an `EnvironmentDefinitionRevision`.

Valid state values:

- `A` (SX001G:approvedStatus)
- `D` (SX001G:draftStatus)
- `IW` (SX001G:inWorkStatus)
- `C` (SX001G:cancelledStatus)
- `R` (SX001G:reviewedStatus)

Source:

- SX001G

Chapter 7.3.13

Units of functionality - UoF Hardware Element

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.46	Units of functionality - UoF Zone Element
SX001G	Glossary for the S-Series IPS specifications

Chap No./Document No.	Title
SX002D	Common data model for the S-Series IPS specifications

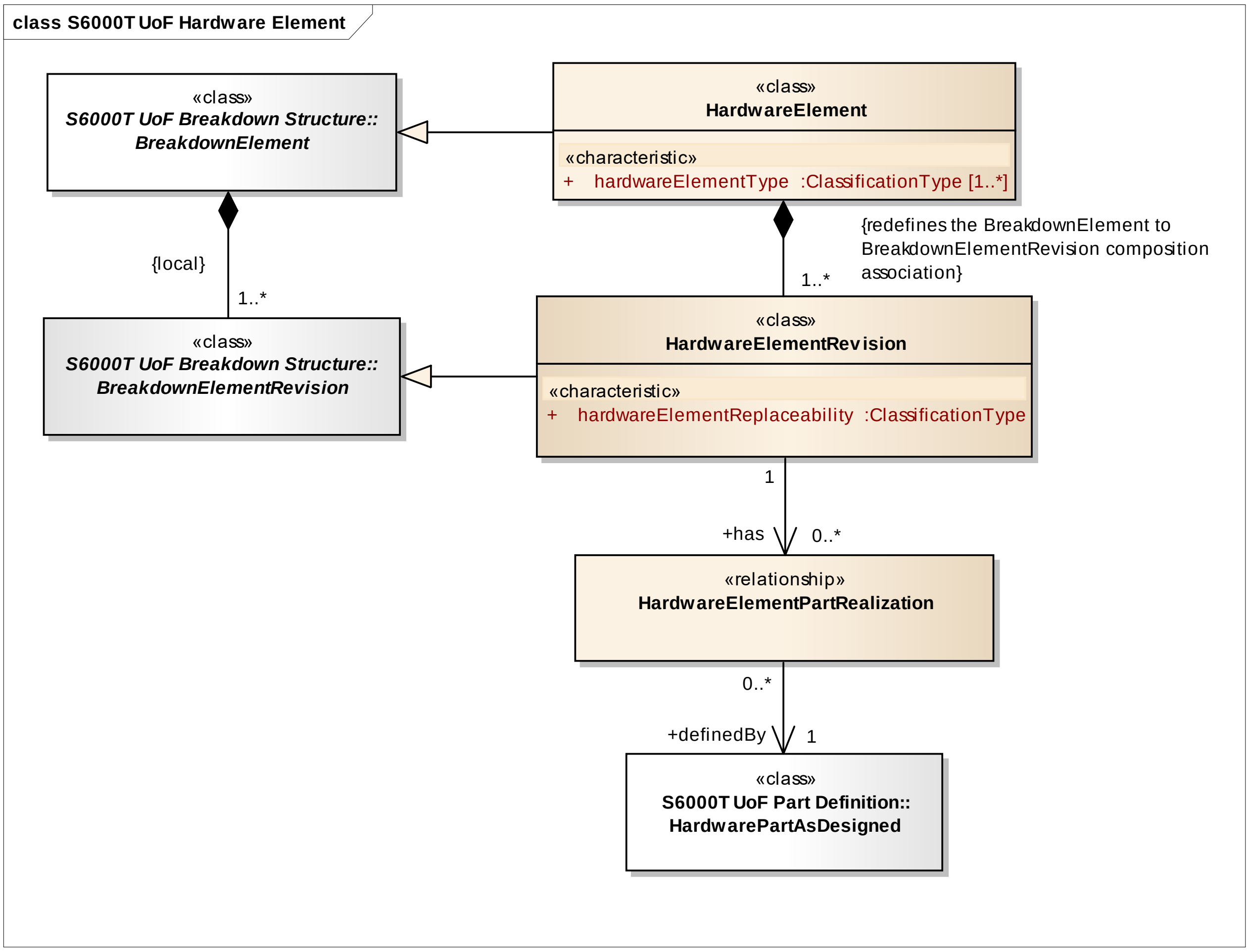
1 General

The Hardware Element UoF provides the capability to specify that an element within a breakdown is hardware and can be associated with the hardware part(s) that fulfill the requirement.

2 UoF Hardware Element

Key features of the UoF Hardware Element data model, (refer to Fig 1), are:

- `HardwareElement` is a specialization of `BreakdownElement` which means that, wherever `BreakdownElement` is being used in the data model, `HardwareElement` can be used instead
- `HardwareElementRevision` is a specialization of `BreakdownElementRevision` which means that, wherever `BreakdownElementRevision` is being used in the data model, `HardwareElementRevision` can be used instead
- an instance of `HardwareElementRevision` can be associated with one or many instances of `HardwarePartAsDesigned` that meet the requirements and specification for the `HardwareElement`



ICN-S6000T-B6865-00030-002-01

Fig 1 UoF Hardware Element

2.1 HardwareElement

2.1.1 Class definition

`HardwareElement` is a `BreakdownElement` (refer to Chap 7.3.3) that is realized as a `HardwarePartAsDesigned`.

Source:

– SX001G

2.1.1.1 Associations

This class has the following associations:

– An aggregate association with one or many instances of `HardwareElementRevision`

2.1.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

– `BreakdownElementInZoneItem` (inherited from `BreakdownElement`). Refer to Chap 7.3.46.
– `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.

- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from BreakdownElement). Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 breakdownElementIdentifier (one or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.1.2.2 breakdownElementName (zero, one or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.1.2.3 breakdownElementEssentiality (zero or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.1.2.4 hardwareElementType (one or many)
hardwareElementType is a classification that identifies further specialization for a HardwareElement.

Examples:

- access point
- door
- electrical panel
- equipment
- panel
- slot

Valid values:

- EQP (SX001G:equipmentHardwareElement)
- OPN (SX001G:openingHardwareElement)
- PNL (SX001G:panelHardwareElement)

Source:

- SX001G

2.2 HardwareElementPartRealization

2.2.1 Class definition

HardwareElementPartRealization is a <<relationship>> where a HardwareElementRevision relates to an instance of HardwarePartAsDesigned (refer to Chap 7.3.21) which fulfills the HardwareElement specification.

Source:

- SX001G

2.2.1.1 Associations
This class has the following associations:

- A directed definedBy association with one instance of HardwarePartAsDesigned. Refer to Chap 7.3.21.

2.2.1.2 Implementations
This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3 HardwareElementRevision

2.3.1 Class definition

HardwareElementRevision is a BreakdownElementRevision (refer to Chap 7.3.3) representing an iteration applied to a HardwareElement.

Source:

- SX001G

2.3.1.1 Associations
This class has the following associations:

- A directed has association with zero, one or many instances of HardwareElementPartRealization

2.3.1.2 Implementations
This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem (inherited from BreakdownElementRevision). Refer to Chap 7.3.46.
- ChangeControlledItem (inherited from BreakdownElementRevision). Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 breakdownElementRevisionIdentifier
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.3.2.2 breakdownElementDescription (zero, one or many)
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.3.2.3 breakdownElementRevisionRationale (zero, one or many)
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.3.2.4 breakdownElementRevisionDate (zero or one)
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.3.2.5 breakdownElementRevisionStatus (zero or one)
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.3.2.6 hardwareElementReplaceability

`hardwareElementReplaceability` is a classification that identifies whether its realization is expected to be replaceable from a technical standpoint, independent from customer maintenance concepts.

Valid values:

- `N` (SX001G:nonReplaceableHardwareElement)
- `R` (SX001G:replaceableHardwareElement)

Source:

- SX001G

Chapter 7.3.14

Units of functionality - UoF Job Duty

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification

Chap No./Document No.	Title
Chap 7.3.32	Units of functionality - UoF Task
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

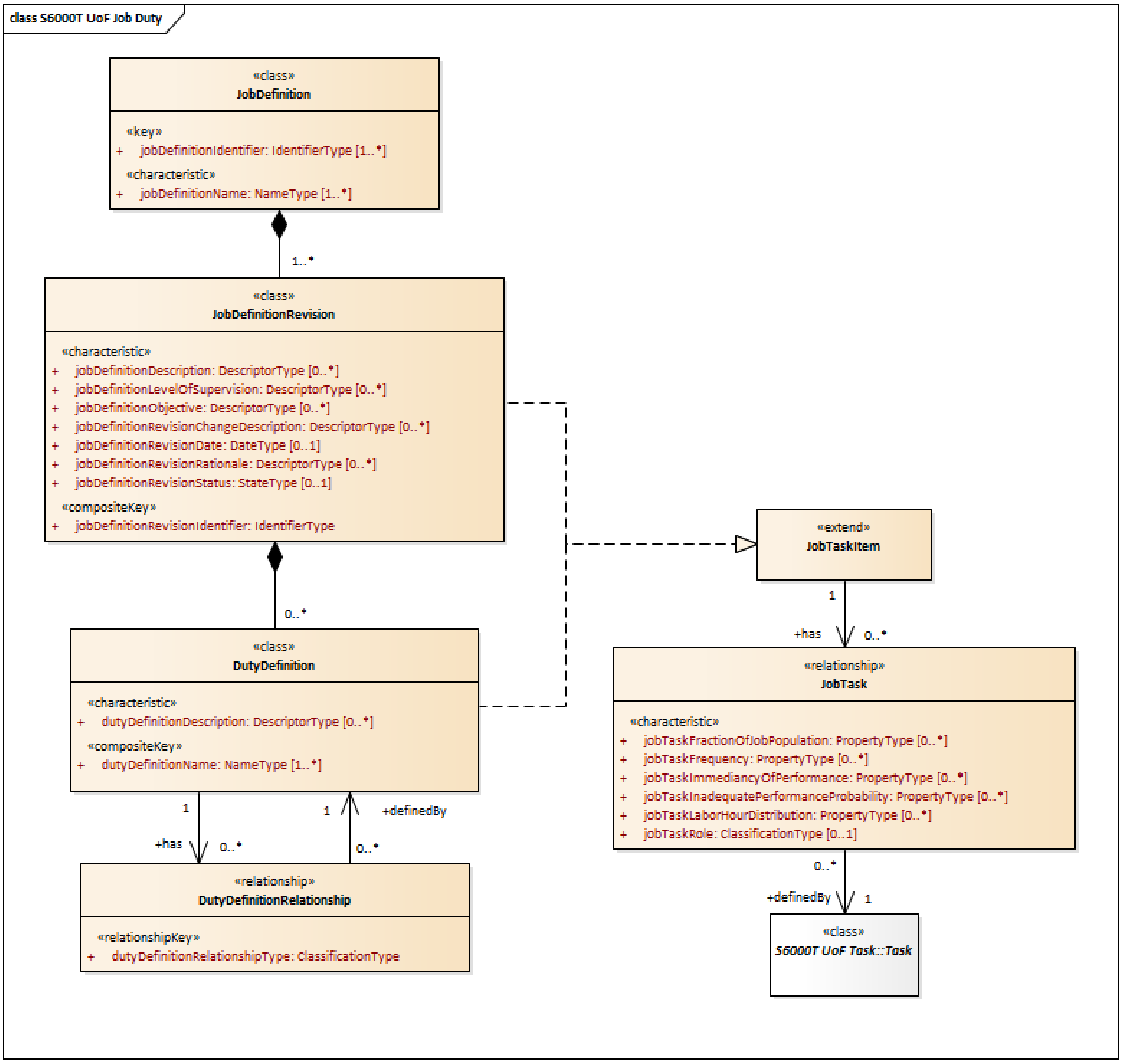
1 General

The Job Duty UoF supports the definition of a position of regular employment, responsibility domains, and specific tasks that need to be carried out on the Product for the overall organization to be successful in reaching its objectives.

2 UoF Job Duty

Key features of the UoF Job Duty data model, (refer to Fig 1), are:

- the definition of a `JobDefinition` can be iterated using `JobDefinitionRevision`
- a `DutyDefinition` can be associated with other `DutyDefinition`s eg, to form a hierarchical structure of duties
- both a `JobDefinition` and a `DutyDefinition` can be associated with a set of `Task`s. For each associated `Task` can a set of job-related characteristics be defined eg, how often will a person in a particular job or duty carry out the associated `Task`.



ICN-S6000T-B6865-00041-003-01

Fig 1 UoF Job Duty

2.1 DutyDefinition

2.1.1 Class definition

DutyDefinition is a <<class>> that defines a functional responsibility within a defined JobDefinition.

Examples:

- aviation technician duties includes perform engine scheduled maintenance
- security guard duties includes eg, perform plant inspections and perform monitor surveillance

2.1.1.1 Associations

This class has the following associations:

- A directed `has` association with zero, one or many instances of `DutyDefinitionRelationship`

2.1.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `DigitalFileReferencingItem`. Refer to Chap 7.3.10.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `JobTaskItem`
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 dutyDefinitionName (one or many)

`dutyDefinitionName` is a name by which the `DutyDefinition` is known and can be easily referenced.

2.1.2.2 dutyDefinitionDescription (zero, one or many)

`dutyDefinitionDescription` is a description that gives more information about the activities that are included in the defined responsibility.

2.2 DutyDefinitionRelationship

2.2.1 Class definition

`DutyDefinitionRelationship` is a `<<relationship>>` where one `DutyDefinition` relates to another `DutyDefinition`.

2.2.1.1 Associations

This class has the following associations:

- A directed `definedBy` association with zero, one or many instances of `DutyDefinition`

2.2.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 dutyDefinitionRelationshipType

`dutyDefinitionRelationshipType` is a classification that identifies the meaning of the established relationship.

Example:

- includes

Valid values:

- INCL (SX001G:includedDutyDefinition)

2.3 JobDefinition

2.3.1 Class definition

JobDefinition is a <<class>> that represents the definitional information for a collection of homogeneous tasks related by similarity of knowledge and skill requirements and can be assigned to a person.

Examples:

- aviation technician
- security guard

2.3.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of JobDefinitionRevision
- An aggregate association with zero, one or many instances of OccupationalBackgrpund. Refer to Chap 7.3.31.
- An aggregate association with zero, one or many instances of TrainingEntryRequirement. Refer to Chap 7.3.39.

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- CurriculumDefinitionTargetAudienceItem. Refer to Chap 7.3.9.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.3.2 Attributes

2.3.2.1 jobDefinitionIdentifier (one or many)

jobDefinitionIdentifier is an identifier that establishes a unique designator for a JobDefinition and to differentiate it from other instances of JobDefinition.

Valid identifier class values:

- ID (SX001G:jobDefinitionRevisionIdentifier)

2.3.2.2 jobDefinitionName (one or many)

jobDefinitionName is a name by which the JobDefinition is known and can be easily referenced.

2.4 JobDefinitionRevision

2.4.1 Class definition

JobDefinitionRevision is a <<class>> representing an iteration applied to a JobDefinition.

2.4.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of DutyDefinition

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- JobTaskItem
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 jobDefinitionRevisionIdentifier

jobDefinitionRevisionIdentifier is an identifier that establishes a unique designator for a JobDefinitionRevision and to differentiate it from other instances of JobDefinitionRevision.

Valid identifier class values:

- ID (SX001G:jobDefinitionRevisionIdentifier)

2.4.2.2 jobDefinitionDescription (zero, one or many)

jobDefinitionDescription is a description that gives more information about the job and its overall tasks and duties.

2.4.2.3 jobDefinitionObjective (zero, one or many)

jobDefinitionObjective is a description that specifies the purpose, responsibilities and expected performance of a person performing the job.

2.4.2.4 jobDefinitionLevelOfSupervision (zero, one or many)

jobDefinitionLevelOfSupervision is a description that defines the degree to which a person performing the job must be watched and directed to make certain that the job is done correctly and safely.

2.4.2.5 jobDefinitionRevisionChangeDescription (zero, one or many)

jobDefinitionRevisionChangeDescription is a description that gives more information about the content that has been altered between two revisions of a job definition.

2.4.2.6 jobDefinitionRevisionRationale (zero, one or many)

jobDefinitionRevisionRationale is a description that gives more information about the justification for revising the JobDefinition

2.4.2.7 jobDefinitionRevisionDate (zero or one)
jobDefinitionRevisionDate is a date that specifies when a JobDefinitionRevision was defined.

2.4.2.8 jobDefinitionRevisionStatus (zero or one)
jobDefinitionRevisionStatus is a state that identifies the maturity of the JobDefinitionRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

2.5 JobTask

2.5.1 Class definition

JobTask is a <<relationship>> that defines an association between a defined JobDefinition or DutyDefinition and a task that provides detailed information on how the job or duty, or portions thereof, is to be carried out.

2.5.1.1 Associations
This class has the following associations:

- A directed definedBy association with one instance of Task. Refer to Chap 7.3.32.

2.5.1.2 Implementations
This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 jobTaskRole (zero or one)
jobTaskRole is a classification that specifies how the JobDefinition or DutyDefinition is related to the performance of the Task. Refer to Chap 7.3.32.

Valid values:

- A (SX001G:assistantJobTaskRole)
- C (SX001G:co-performerJobTaskRole)
- P (SX001G:performerJobTaskRole)

2.5.2.2 jobTaskFrequency (zero, one or many)
jobTaskFrequency is a property that specifies the expected rate of occurrence of a Task (refer to Chap 7.3.32) in the context of a specific JobDefinition.

Valid unit value:

– timeCycleRateUnit

2.5.2.3 jobTaskInadequatePerformanceProbability (zero, one or many)
`jobTaskInadequatePerformanceProbability` is a property that specifies the likelihood of unsatisfactory completion of a `Task` (refer to Chap 7.3.32) in the context of a specific `JobDefinition`.

Valid unit value:

– relativeUnit

2.5.2.4 jobTaskLaborHourDistribution (zero, one or many)
`jobTaskLaborHourDistribution` is a property that specifies the time spent on the associated `Task` (refer to Chap 7.3.32) relative to all other `Task`s and commitments that are associated with the `JobDefinition`.

Valid unit value:

– relativeUnit

2.5.2.5 jobTaskImmediancyOfPerformance
`jobTaskImmediancyOfPerformance` is a property that specifies the expected time between training and job performance for a `Task` (refer to Chap 7.3.32) in the context of a the `JobDefinition`.

Valid unit value:

– timeUnit

2.5.2.1 jobTaskFractionOfJobPopulation
`jobTaskFractionOfJobPopulation` is a property property that specifies the fraction of the Job population that will perform the associated Task.

2.6 JobTaskItem

2.6.1 Interface definition

`JobTaskItem` is an `<<extend>>` interface that provides its associated data model to those classes that implement it.

2.6.1.1 Associations
The `JobTaskItem` `<<extend>>` interface has the following associations:

– A directed `has` association with zero, one or many instances of `JobTask`

2.6.1.2 Class members
Classes that implement the `JobTaskItem` `<<extend>>` interface are:

– `DutyDefinition`
– `JobDefinitionRevision`

Chapter 7.3.15

Units of functionality - UoF Learning Assessment

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.11	Units of functionality - UoF Task resource competence definition
Chap 7.3.17	Units of functionality - UoF Task train prioritization
Chap 7.3.26	Units of functionality - UoF Instructional strategy
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

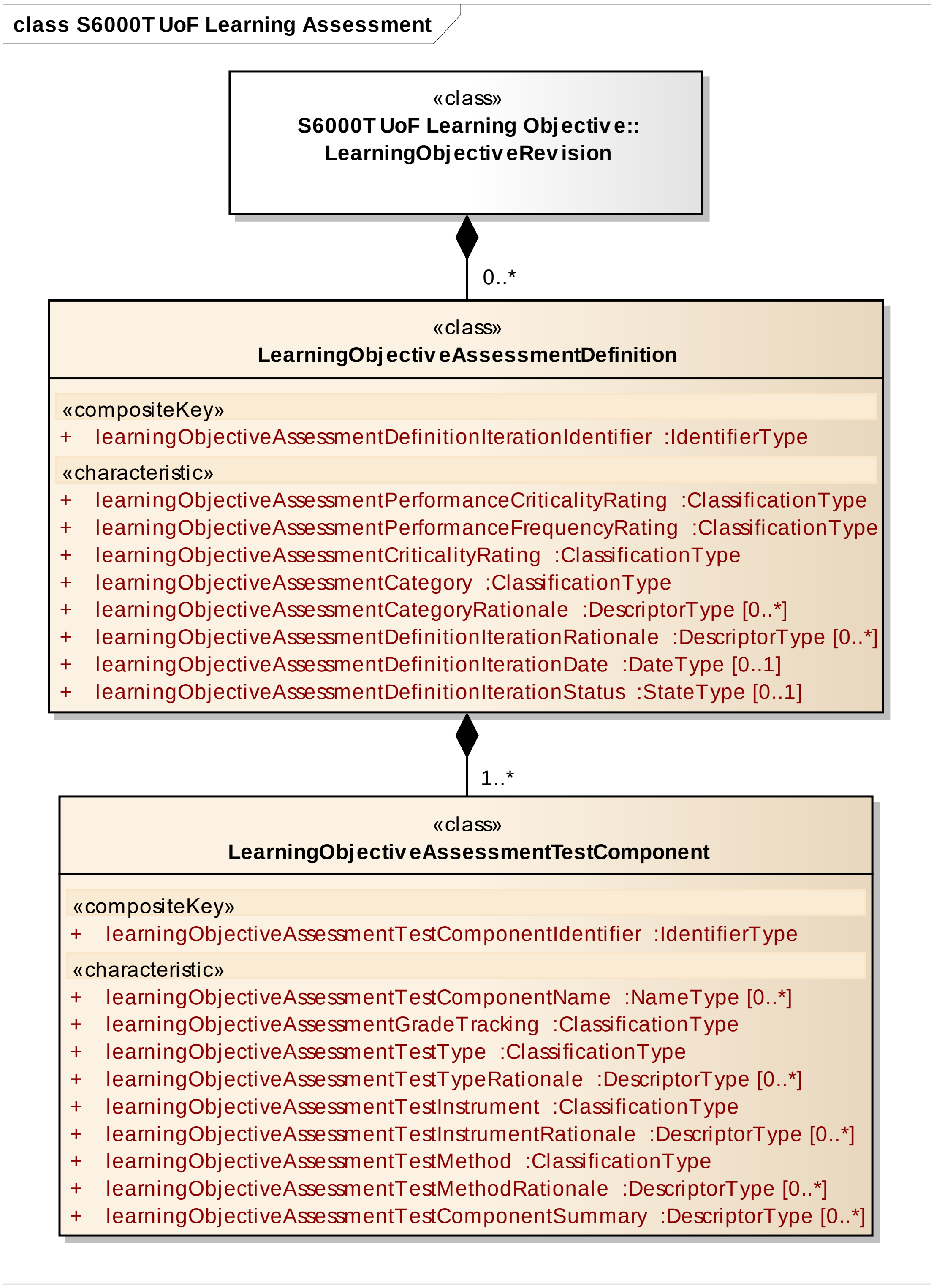
The Learning Assessment UoF provides the capability to document requirements for measuring and asserting the target audience's attainment of learning objectives.

2 UoF Learning Assessment

Key features of the UoF Learning Assessment data model, (refer to Fig 1), are:

- the requirements for measuring and asserting the target audience's attainment of learning is defined per learning objective

- the definition of requirements for measuring and asserting the target audience's attainment of learning can be iterated for an individual learning objective
- the learning assessment for an individual learning objective can include one or many test components



ICN-S6000T-B6865-00051-002-01

Fig 1 UoF Learning Assessment

2.1 LearningObjectiveAssessmentDefinition

2.1.1 Class definition

LearningObjectiveAssessmentDefinition is a <<class>> that specifies requirements for measuring and asserting the target audience's attainment of the associated LearningObjective (refer to Chap 7.3.17).

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of LearningObjectiveAssessmentTestComponent

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 learningObjectiveAssessmentDefinitionIterationIdentifier

learningObjectiveAssessmentDefinitionIterationIdentifier is an identifier that establishes a unique designator for the LearningObjectiveAssessmentDefinition and to differentiate it from other instances of LearningObjectiveAssessmentDefinition.

Valid identifier class values:

- ID (SX001G:learningObjectiveAssessmentDefinitionIterationIdentifier)

2.1.2.2 learningObjectiveAssessmentPerformanceCriticalityRating

learningObjectiveAssessmentPerformanceCriticalityRating is a classification that defines the need for assessing the LearningObjective (refer to Chap 7.3.17) relative to how essential the defined knowledge, skills and attitude is to competent task performance.

Valid values:

- H (SX001G:highLearningObjectiveAssessmentPerformanceCriticality)
- L (SX001G:lowLearningObjectiveAssessmentPerformanceCriticality)
- M (SX001G:mediumLearningObjectiveAssessmentPerformanceCriticality)

2.1.2.3 learningObjectiveAssessmentPerformanceFrequencyRating

learningObjectiveAssessmentPerformanceFrequencyRating is a classification that defines the importance of testing the LearningObjective (refer to Chap 7.3.17) relative to how often the knowledge, skills or attitude is used during task performance.

Valid values:

- H (SX001G:highLearningObjectivePerformanceFrequency)
- L (SX001G:lowLearningObjectivePerformanceFrequency)
- M (SX001G:mediumLearningObjectivePerformanceFrequency)

2.1.2.4 learningObjectiveAssessmentCriticalityRating

`learningObjectiveAssessmentCriticalityRating` is a classification that determines the importance of formally testing a `LearningObjective` (refer to Chap 7.3.17).

Valid values:

- `H` (SX001G:highLearningObjectiveAssessmentCriticality)
- `L` (SX001G:lowLearningObjectiveAssessmentCriticality)
- `M` (SX001G:mediumLearningObjectiveAssessmentCriticality)

2.1.2.5 learningObjectiveAssessmentCategory

`learningObjectiveAssessmentCategory` is a classification that identifies the overall test approach for measuring and asserting the target audience's attainment of the `LearningObjective` (refer to Chap 7.3.17).

Valid values:

- `K` SX001G:knowledgeBasedLearningObjectiveAssessment
- `P` SX001G:performanceBasedLearningObjectiveAssessment

2.1.2.6 learningObjectiveAssessmentCategoryRationale (zero, one or many)

`learningObjectiveAssessmentCategoryRationale` is a description that provides information about the reason for the selected `learningObjectiveAssessmentCategory`.

2.1.2.7 learningObjectiveAssessmentDefinitionIterationRationale (zero, one or many)

`learningObjectiveAssessmentDefinitionIterationRationale` is a description that gives more information about the justification for iterating the `LearningObjectiveAssessmentDefinition`.

2.1.2.8 learningObjectiveAssessmentDefinitionIterationDate (zero or one)

`learningObjectiveAssessmentDefinitionIterationDate` is a date that specifies when the decision on assessment strategy for the `LearningObjective` (refer to Chap 7.3.17) was iterated.

2.1.2.9 learningObjectiveAssessmentDefinitionIterationStatus (zero or one)

`learningObjectiveAssessmentDefinitionIterationStatus` is a state that identifies the maturity of the `LearningObjectiveAssessmentDefinition`.

Valid state values:

- `A` (SX001G:approvedStatus)
- `D` (SX001G:draftStatus)
- `IW` (SX001G:inWorkStatus)
- `C` (SX001G:cancelledStatus)
- `R` (SX001G:reviewedStatus)

2.2 LearningObjectiveAssessmentTestComponent

2.2.1 Class definition

`LearningObjectiveAssessmentTestComponent` is a `<<class>>` that defines a specific evaluation and measurement of the target audience's attainment of the associated `LearningObjective` or portion thereof.

2.2.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 learningObjectiveAssessmentTestComponentIdentifier

learningObjectiveAssessmentTestComponentIdentifier is an identifier that establishes a unique designator for the LearningObjectiveAssessmentTestComponent and to differentiate it from other instances of LearningObjectiveAssessmentTestComponent.

Valid identifier class values:

- ID (SX001G:learningObjectiveAssessmentTestComponentIdentifier)

2.2.2.2 learningObjectiveAssessmentTestComponentName (zero, one or many)

learningObjectiveAssessmentTestComponentName is a name by which the LearningObjectiveAssessmentTestComponent is known and can be easily referenced.

2.2.2.3 learningObjectiveAssessmentGradeTracking

learningObjectiveAssessmentGradeTracking is a classification that defines whether the achievement of the LearningObjective (refer to Chap 7.3.17) or portion thereof is recorded as part of a trainee's grade.

Valid values:

- F (SX001G:formalLearningObjectiveAssessment)
- I (SX001G:informalLearningObjectiveAssessment)

2.2.2.4 learningObjectiveAssessmentTestType

learningObjectiveAssessmentTestType is a classification that defines the placement and purpose of a given LearningObjectiveAssessmentTestComponent within the overall assessment strategy.

Valid values:

- COMP (SX001G:comprehensiveTestLearningObjectiveAssessmentTestComponent)
- DIAG (SX001G:diagnosticLearningObjectiveAssessmentTestComponent)
- ENTR (SX001G:entryLearningObjectiveAssessmentTestComponent)
- FORM (SX001G:formativeLearningObjectiveAssessmentTestComponent)
- PRE (SX001G:preTestLearningObjectiveAssessmentTestComponent)
- PROG (SX001G:progressLearningObjectiveAssessmentTestComponent)
- SUMM (SX001G:summativeTestLearningObjectiveAssessmentTestComponent)

2.2.2.5 learningObjectiveAssessmentTestTypeRationale (zero, one or many)

`learningObjectiveAssessmentTestTypeRationale` is a description that gives more information on the justification for selecting the `learningObjectiveAssessmentTestType`.

2.2.2.6 learningObjectiveAssessmentTestInstrument

`learningObjectiveAssessmentTestInstrument` is a classification that defines the type of instrument that is to be used for measuring the target audience's attainment of the associated `LearningObjective` (refer to Chap 7.3.17) or portion thereof.

Valid values:

- `K-CMPL` (SX001G:competionKnowledgeTestInstrument)
- `K-MC` (SX001G:multipleChoiceKnowledgeTestInstrument)
- `K-MS` (SX001G:multipleSelectionKnowledgeTestInstrument)
- `K-MTCH` (SX001G:matchingKnowledgeTestInstrument)
- `K-TF` (SX001G:trueFalseKnowledgeTestInstrument)
- `P-CHL` (SX001G:checklistPerformanceTestInstrument)
- `P-JS` (SX001G:jobSheetPerformanceTestInstrument)
- `P-PROC` (SX001G:procedurePerformanceTestInstrument)

2.2.2.7 learningObjectiveAssessmentTestInstrumentRationale (zero, one or many)

`learningObjectiveAssessmentTestInstrumentRationale` is a description that gives more information on the justification for selecting the `learningObjectiveAssessmentTestInstrument`.

2.2.2.8 learningObjectiveAssessmentTestMethod

`learningObjectiveAssessmentTestMethod` is a classification that defines the scoring method that will be used when testing achievement of the associated `LearningObjective` (refer to Chap 7.3.17) or portion thereof.

Valid values:

- `CRIT` (SX001G:criterionReferencedLearningObjectiveAssessmentTestMethod)
- `NORM` (SX001G:normReferencedLearningObjectiveAssessmentTestMethod)

2.2.2.9 learningObjectiveAssessmentTestMethodRationale (zero, one or many)

`learningObjectiveAssessmentTestMethodRationale` is a description that gives more information on the justification for selecting the `learningObjectiveAssessmentTestMethod`.

2.2.2.10 learningObjectiveAssessmentTestComponentSummary (zero, one or many)

`learningObjectiveAssessmentTestComponentSummary` is a description that provides summary of the defined `LearningObjectiveAssessmentTestComponent` including possible important and unique characteristics.

Chapter 7.3.16

Units of functionality - UoF Learning Gap

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

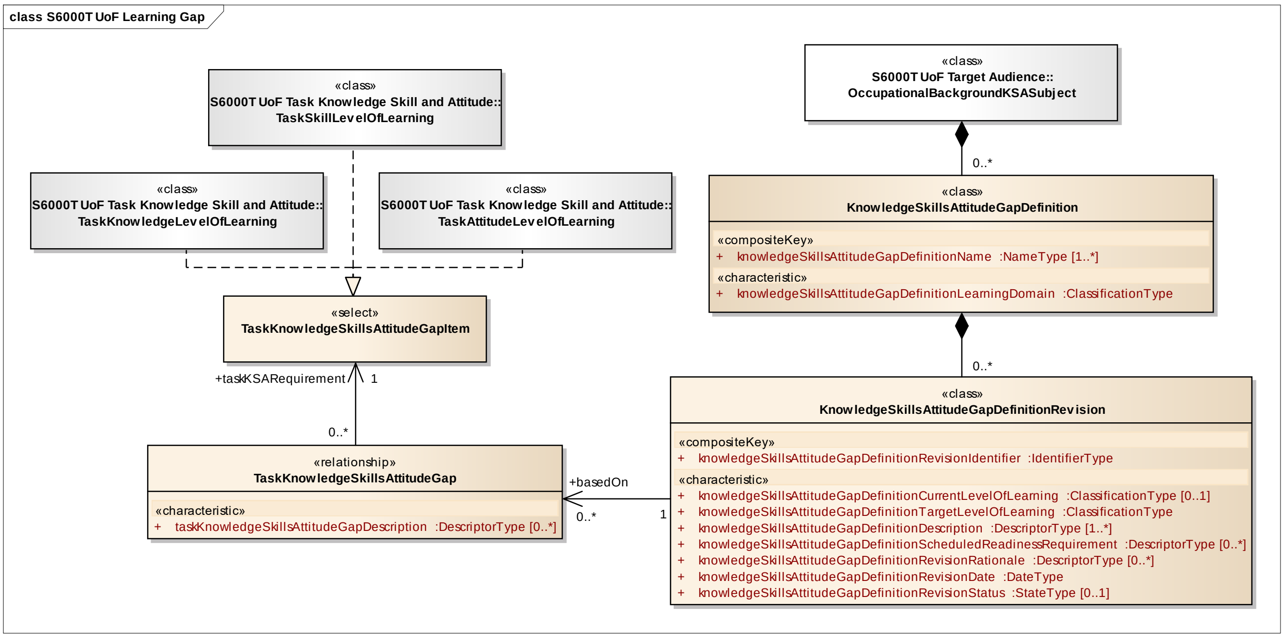
1 General

The Learning Gap UoF provides the capability to define gaps in knowledge, skills and attitude for the target audience compared to the knowledge, skills and attitudes that is required for the performance of a task.

2 UoF Learning Gap

Key features of the UoF Learning Gap data model, (refer to Fig 1), are:

- Knowledge Skill and Attitude (KSA) gaps are defined against occupational background KSA subjects defined for the target audience. Refer to Chap 7.3.31.
- KSA gaps can refer to one or many KSA requirements identified for the tasks to be carried out



ICN-S6000T-B6865-00049-002-01

Fig 1 UoF Learning Gap

2.1 KnowledgeSkillsAttitudeGapDefinition

2.1.1 Class definition

`KnowledgeSkillsAttitudeGapDefinition` is a `<<class>>` that defines a deficiency in knowledge, skills or attitude that must be addressed for the identified target audience.

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `KnowledgeSkillsAttitudeGapDefinitionRevision`

2.1.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.

– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 knowledgeSkillsAttitudeGapDefinitionName (one or many)

knowledgeSkillsAttitudeGapDefinitionName is a name by which the KnowledgeSkillsAttitudeGapDefinition is known and can be easily referenced.

2.1.2.2 knowledgeSkillsAttitudeGapDefinitionLearningDomain

knowledgeSkillsAttitudeGapDefinitionLearningDomain is a classification that determines to which learning domain the gap is written.

Valid values:

- A (SX001G:attitudeGapDefinition)
- K (SX001G:knowledgeGapDefinition)
- S (SX001G:skillGapDefinition)

2.2 KnowledgeSkillsAttitudeGapDefinitionRevision

2.2.1 Class definition

KnowledgeSkillsAttitudeGapDefinitionRevision is a <<class>> representing an iteration applied to a KnowledgeSkillsAttitudeGapDefinition.

2.2.1.1 Associations

This class has the following associations:

- A directed basedOn association with zero, one or many instances of TaskKnowledgeSkillsAttitudeGap

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 knowledgeSkillsAttitudeGapDefinitionRevisionIdentifier

knowledgeSkillsAttitudeGapDefinitionRevisionIdentifier is an identifier that establishes a unique designator for a KnowledgeSkillsAttitudeGapDefinitionRevision and to differentiate it from other instances of KnowledgeSkillsAttitudeGapDefinitionRevision.

Valid identifier class values:

- ID (SX001G:knowledgeSkillsAttitudeGapDefinitionRevisionIdentifier)

2.2.2.2 knowledgeSkillsAttitudeGapDefinitionCurrentLevelOfLearning (zero or one)

knowledgeSkillsAttitudeGapDefinitionCurrentLevelOfLearning is a classification that identifies the target audience estimated level of sophistication for the defined learning domain.

Note

Current level of learning is just required where the gap is defined at a more detailed level then what is defined for the occupational background KSA subject and its associated rating.

Valid values:

– Refer to the S6000T valid values XML schema for details

2.2.2.3 knowledgeSkillsAttitudeGapDefinitionTargetLevelOfLearning

`knowledgeSkillsAttitudeGapDefinitionTargetLevelOfLearning` is a classification that identifies the required level of sophistication for the defined learning domain.

Note

Target level of learning summarizes the required level of learning defined for the associated task KSA requirements.

Valid values:

– Refer to the S6000T valid values XML schema for details

2.2.2.4 knowledgeSkillsAttitudeGapDefinitionDescription (one or many)

`knowledgeSkillsAttitudeGapDefinitionDescription` is a description that gives more information on the identified deficiency in knowledge, skills or attitude.

2.2.2.5 knowledgeSkillsAttitudeGapDefinitionScheduledReadinessRequirement (zero, one or many)

`knowledgeSkillsAttitudeGapDefinitionScheduledReadinessRequirement` is a description that specifies when the target audience must have the full operational capability.

Note

The point of reference for when the target audience must have the full operational capability can be defined relative to the product introduction, the introduction of operational capabilities, planned missions, planned deployments, or just be a date.

2.2.2.6 knowledgeSkillsAttitudeGapDefinitionRevisionRationale (zero, one or many)

`knowledgeSkillsAttitudeGapDefinitionRevisionRationale` is a description that gives more information on the justification for revising the `KnowledgeSkillsAttitudeGapDefinition`.

2.2.2.7 knowledgeSkillsAttitudeGapDefinitionRevisionDate

`knowledgeSkillsAttitudeGapDefinitionRevisionDate` is a date that specifies when the `KnowledgeSkillsAttitudeGapDefinition` was revised.

2.2.2.8 knowledgeSkillsAttitudeGapDefinitionRevisionStatus (zero or one)

`knowledgeSkillsAttitudeGapDefinitionRevisionStatus` is a state that identifies the maturity of the `KnowledgeSkillsAttitudeGapDefinitionRevision`.

Valid state values:

– `A` (SX001G:approvedStatus)
– `D` (SX001G:draftStatus)
– `IW` (SX001G:inWorkStatus)
– `C` (SX001G:cancelledStatus)
– `R` (SX001G:reviewedStatus)

2.3 TaskKnowledgeSkillsAttitudeGap

2.3.1 Class definition

TaskKnowledgeSkillsAttitudeGap is a <<relationship>> that identifies the task context for which the deficiency in qualification has been identified.

2.3.1.1 Associations

This class has the following associations:

- A directed taskKSARequirement association with one instance of TaskKnowledgeSkillsAttitudeGapItem

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 taskKnowledgeSkillsAttitudeGapDescription (zero, one or many)

taskKnowledgeSkillsAttitudeGapDescription is a description that gives more information on the identified deficiency in qualifications in relation to the required knowledge, skills or attitude defined for a Task (refer to Chap 7.3.32), or portion thereof.

2.4 TaskKnowledgeSkillsAttitudeGapItem

2.4.1 Interface definition

TaskKnowledgeSkillsAttitudeGapItem is a <<select>> interface that identifies items that represent qualification requirements.

2.4.1.1.1 Class members

This <<select>> interface includes the following class members:

- TaskAttitudeLevelOfLearning. Refer to Chap 7.3.33.
- TaskKnowledgeLevelOfLearning. Refer to Chap 7.3.33.
- TaskSkillLevelOfLearning. Refer to Chap 7.3.33.

Chapter 7.3.17

Units of functionality - UoF Learning Objective

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 4.3	Design - Learning gaps
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.4	Units of functionality - UoF Hardware element
Chap 7.3.10	Units of functionality - UoF Digital File

Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.15	Units of functionality - UoF Learning Assessment
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.4.41	Units of functionality - UoF Training Media and Fidelity
Chap 7.4.43	Units of functionality - UoF Training Method
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

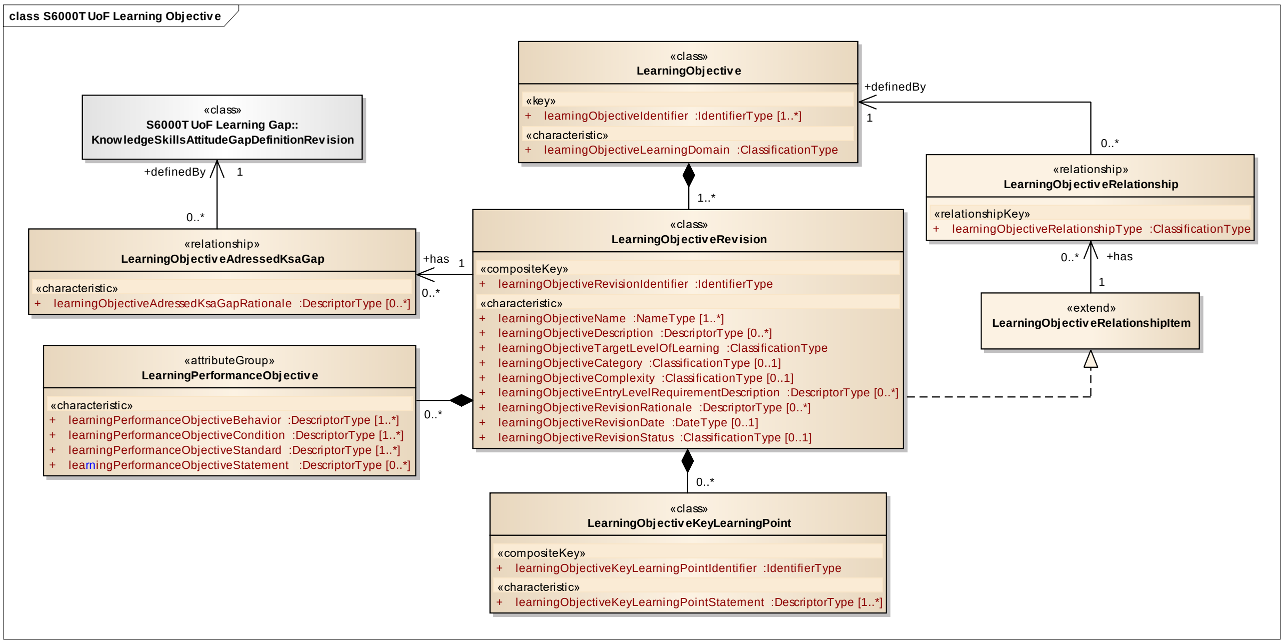
1 General

The Learning Objective UoF supports the definition of expected outcome of a learning activity including what trainees must demonstrate and the conditions under which they will demonstrate the acquired knowledge, skills or attitude.

2 UoF Learning Objective

Key features of the UoF Learning Objective data model, (refer to Fig 1), are:

- Learning Objectives (LO) can be associated with Knowledge, Skills or Attitude (KSA) gaps identified during learning gaps analysis (refer to Chap 4.3)
- LO can have a set of defined key learning points
- the definition of learning objectives can be iterated using `LearningObjectiveRevision`
- LO can be associated with other learning objectives to define eg, hierarchies of learning objectives



ICN-S6000T-B6865-00050-002-01

Fig 1 UoF Learning Objective

2.1 LearningObjective

2.1.1 Class definition

LearningObjective is a <<class>> that defines the expected outcome of a learning activity in terms of assessable knowledge, skills or attitude that will be acquired by a trainee as a result of instruction.

Note

LO when attained should be observable behaviors or actions.

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of LearningObjectiveRevision

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 learningObjectiveIdentifier (one or many)

learningObjectiveIdentifier is an identifier that establishes a unique designator for a LearningObjective and to differentiate it from other instances of LearningObjective.

Valid identifier class values:

- ID (SX001G:learningObjectiveIdentifier)

2.1.2.2 learningObjectiveLearningDomain

learningObjectiveLearningDomain is a classification that determines the learning aspect for which the LearningObjective is written.

Valid values:

- A (SX001G:attitudeLearningDomain)
- K (SX001G:knowledgeLearningDomain)
- N/A (SX001G:notApplicable)
- S (SX001G:skillLearningDomain)

2.2 LearningObjectiveAdressedKsaGap

2.2.1 Class definition

LearningObjectiveAdressedKsaGap is a <<relationship>> that identifies a gap in the target audience knowledge, skills or attitude which the associated LearningObjective addresses.

2.2.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of KnowledgeSkillsAttitudeGapDefinitionRevision (refer to Chap 7.3.16)

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 learningObjectiveAdressedKsaGapRationale (zero, one or many)

learningObjectiveAdressedKsaGapRationale is a description that gives more information on the justification and extent to which the identified LearningObjective addresses the identified gap in knowledge, skill or attitude.

2.3 LearningObjectiveKeyLearningPoint

2.3.1 Class definition

LearningObjectiveKeyLearningPoint is a <<class>> that defines a specific detail, aspect or quality that is important in the context of the LearningObjective.

2.3.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 learningObjectiveKeyLearningPointIdentifier

learningObjectiveKeyLearningPointIdentifier is an identifier that establishes a unique designator for a LearningObjectiveKeyLearningPoint and to differentiate it from other instances of LearningObjectiveKeyLearningPoint.

Valid identifier class values:

- ID (SX001G:learningObjectiveKeyLearningPointIdentifier)

2.3.2.2 learningObjectiveKeyLearningPointStatement (one or many)

learningObjectiveKeyLearningPointStatement is a description that gives more information on a specific aspect of the knowledge, skills or attitude that must be learnt for the LearningObjective.

2.4 LearningObjectiveRelationship

2.4.1 Class definition

LearningObjectiveRelationship is a <<relationship>> where one LearningObjective relates to another LearningObjective.

2.4.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of LearningObjective

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 learningObjectiveRelationshipType

learningObjectiveRelationshipType is a classification that identifies the meaning of the established relationship.

Valid values:

- P (SX001G:prerequisiteLearningObjective)
- S (SX001G:supportingLearningObjective)

2.5 LearningObjectiveRelationshipItem

2.5.1 Interface definition

LearningObjectiveRelationshipItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

2.5.1.1 Associations

The LearningObjectiveRelationshipItem <<extend>> interface has the following associations:

- A directed has association with zero, one or many instances of LearningObjectiveRelationship

2.5.1.2 Class members

Classes that implement the LearningObjectiveRelationshipItem <<extend>> interface are:

- LearningObjectiveRevision
- LearningObjectiveTrainingMedia. Refer to Chap 7.3.41.
- LearningObjectiveTrainingMethodIteration. Refer to Chap 7.3.43.

2.6 LearningObjectiveRevision

2.6.1 Class definition

LearningObjectiveRevision is a <<class>> representing an iteration applied to a LearningObjective.

2.6.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of LearningObjectiveAssessmentDefinition. Refer to Chap 7.3.15.

- An aggregate association with zero, one or many instances of LearningObjectiveKeyLearningPoint
- An aggregate association with zero, one or many instances of LearningObjectiveTrainingMethod. Refer to Chap 7.3.43.
- An aggregate association with zero, one or many instances of LearningPerformanceObjective
- A directed has association with zero, one or many instances of LearningObjectiveAdressedKsaGap

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem. Refer to Chap 7.3.4.
- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- LearningObjectiveRelationshipItem
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 learningObjectiveRevisionIdentifier

learningObjectiveRevisionIdentifier is an identifier that establishes a unique designator for a LearningObjectiveRevision and to differentiate it from other instances of LearningObjectiveRevision.

Valid identifier class values:

- ID (SX001G:learningObjectiveRevisionIdentifier)

2.6.2.2 learningObjectiveName (one or many)

learningObjectiveName is a name by which the LearningObjective is known and can be easily referenced.

2.6.2.3 learningObjectiveDescription (zero, one or many)

learningObjectiveDescription is a description that gives more information about the defined LearningObjective.

2.6.2.4 learningObjectiveTargetLevelOfLearning

learningObjectiveTargetLevelOfLearning is a classification that identifies the required level of sophistication for the defined LearningObjective.

Valid values:

- Refer to the S6000T valid values XML schema for details

2.6.2.5 learningObjectiveCategory (zero or one)

learningObjectiveCategory is a classification that identifies further specialization for the LearningObjective.

Valid values:

- ELO (SX001G:enablingLearningObjective)

- TLO (SX001G:terminalLearningObjective)

2.6.2.6 learningObjectiveComplexity (zero or one)
learningObjectiveComplexity is a classification that identifies the span and extent of the content needed to attain the defined knowledge, skill, or attitude.

Valid values:

- L (SX001G:lowLearningObjectiveComplexity)
- M (SX001G:mediumLearningObjectiveComplexity)
- H (SX001G:highLearningObjectiveComplexity)

2.6.2.7 learningObjectiveEntryLevelRequirementDescription (zero, one or many)
learningObjectiveEntryLevelRequirementDescription is a description that gives more information on the knowledge, skills or attitude that is a prerequisite for how the LearningObjective is written.

2.6.2.8 learningObjectiveRevisionRationale (zero, one or many)
learningObjectiveRevisionRationale is a description that gives more information about the justification for revising the LearningObjective.

2.6.2.9 learningObjectiveRevisionDate (zero or one)
learningObjectiveRevisionDate is a date that specifies when the LearningObjective was revised.

2.6.2.10 learningObjectiveRevisionStatus (zero or one)
learningObjectiveRevisionStatus is a state that identifies the maturity of the LearningObjectiveRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

2.7 LearningPerformanceObjective

2.7.1 Attribute group definition

LearningPerformanceObjective is an <<attributeGroup>> that defines the precise statement of performance expected at the end of the learning activity.

2.7.2 Attributes

2.7.2.1 learningPerformanceObjectiveBehavior (one or many)
learningPerformanceObjectiveBehavior is a description of the action that demonstrates that a trainee has acquired the defined knowledge, skill or attitude.

2.7.2.2 learningPerformanceObjectiveCondition (one or many)
learningPerformanceObjectiveCondition is a description that expresses the circumstances under which a trainee demonstrates the defined knowledge, skill or attitude.

2.7.2.3 learningPerformanceObjectiveStandard (one or many)

`learningPerformanceObjectiveStandard` is a statement that describes the criteria to which a trainee will be measured when demonstrating the defined knowledge, skills or attitude.

2.7.2.4 learningPerformanceObjectiveStatement (zero, one or many)

`learningPerformanceObjectiveStatement` is a description that expresses the `LearningObjective` behavior, conditions, and standards in one sentence.

Chapter 7.3.18

Units of functionality - UoF Message

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.20	Units of functionality - UoF Organization
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification

Chap No./Document No.	Title
Chap 7.3.38	Units of functionality - UoF Training Analysis and Design Message Content
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

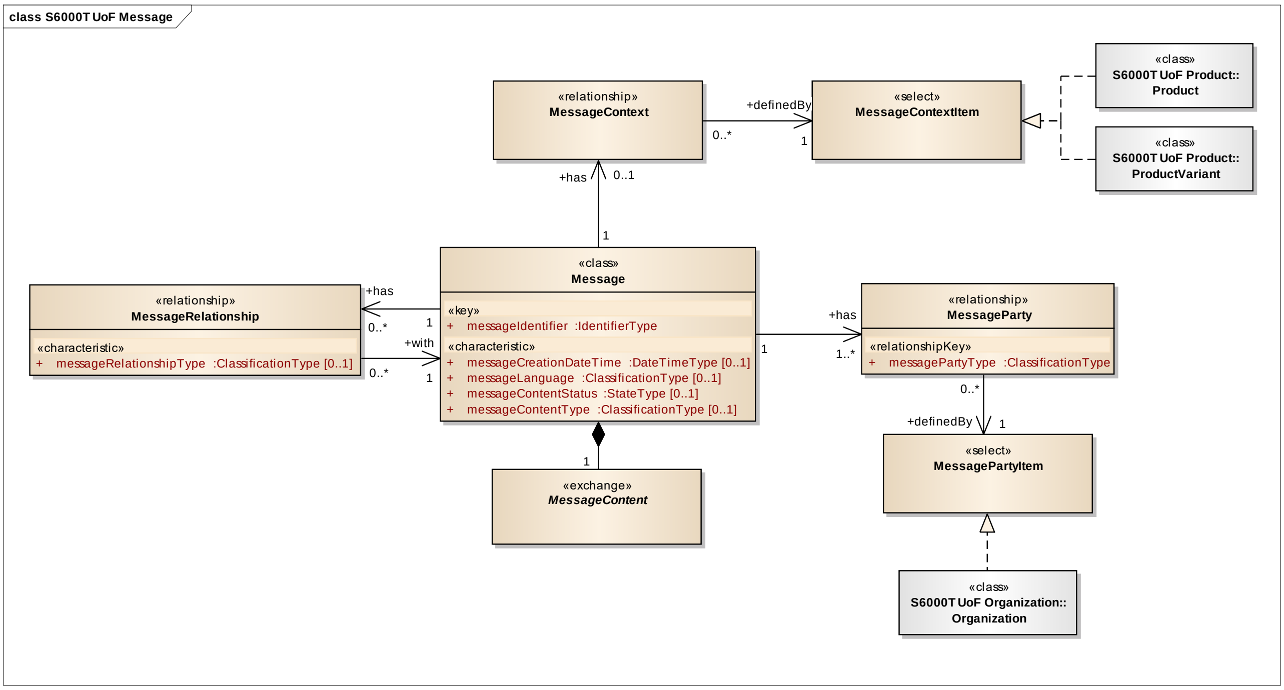
1 General

The Message UoF provides the capability to identify a collection of information to be communicated from one party to another.

2 UoF Message

Key features of the UoF Learning Gap data model, (refer to Fig 1), are:

- message context provides the context within which the message is defined
- message content represent the collection of information that is the subject for the message. Refer to Chap 7.3.38.
- message party identifies the message stakeholders eg, sender and receiver



ICN-S6000T-B6865-00067-002-01

Fig 1 UoF Message

2.1 Message

2.1.1 Class definition

`Message` is a `<<class>>` that represents the collection of information brought together by a message sender for the purpose of communicating it to another party.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with one instance of MessageContent
- A directed has association with one or many instances of MessageParty
- A directed has association with zero or one instance of MessageContext
- A directed has association with zero, one or many instances of MessageRelationship

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 messageIdentifier

messageIdentifier is an identifier that establishes a unique designator for a Message and allows it to be differentiated from other instances of Message.

Source:

- SX001G

2.1.2.2 messageCreationDateTime (zero or one)

messageCreationDateTime is a date and time that defines when the Message was generated.

Source:

- SX001G

2.1.2.3 messageLanguage (zero or one)

messageLanguage is a classification that identifies the language of the information in the message content.

Valid values:

- Refer to the S6000T valid values XML schema for details

Source:

- SX001G

2.1.2.4 messageContentStatus (zero or one)

messageContentStatus is a state that identifies the quality assurance status of the message content.

Valid values:

- D (SX001G:draftMessageContent)
- F (SX001G:finalMessageContent)
- P (SX001G:preliminaryMessageContent)

Source:

- SX001G

2.1.2.5 messageContentType (zero or one)

messageContentType is a classification that characterizes the information included in the message content.

Examples:

- baseline
- net change

Valid values:

- B (SX001G:baselineMessage)
- U (SX001G:netChangeMessage)

Source:

- SX001G

2.2 MessageContent

2.2.1 Class definition

MessageContent is a <<exchange>> definition that represents the collection of information that is the subject of the Message.

Source:

- SX001G

2.3 MessageContext

2.3.1 Class definition

MessageContext is a <<relationship>> between a Message and the context for which it is being provided.

Source:

- SX001G

2.3.1.1 Associations

This class has the following associations:

- A directed has association with one instance of MessageContextItem

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4 MessageContextItem

2.4.1 Interface definition

MessageContextItem is a <<select>> interface that identifies items which can be selected as the context for a Message.

Source:

- SX001G

2.4.1.1 Class members

This <<select>> interface includes the following class members:

- Product. Refer to Chap 7.3.23.
- ProductVariant. Refer to Chap 7.3.23.

2.5 MessageParty

2.5.1 Class definition

MessageParty is a <<relationship>> between a Message and a stakeholder for the Message.

Source:

- SX001G

2.5.1.1 Associations

This class has the following associations:

- A directed association with one instance of MessagePartyItem

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 messagePartyType

messagePartyType is a classification that identifies the role of the associated party.

Examples:

- receiver
- sender

Valid values:

- F (SX001G:messageForwarder)
- R (SX001G:messageReceiver)
- S (SX001G:messageSender)

Source:

- SX001G

2.6 MessagePartyItem
2.6.1 Interface definition
MessagePartyItem is a <<select>> interface that identifies items which can be selected as the party for a Message.

Source:

- SX001G

2.6.1.1 Class members
This <<select>> interface includes the following class members:

- Organization. Refer to Chap 7.3.20.

2.7 MessageRelationship
2.7.1 Class definition
MessageRelationship is a <<relationship>> where one Message relates to another Message.

Examples:

- one Message is a reply to another Message
- one Message is an update to another Message

Source:

- SX001G

2.7.1.1 Associations
This class has the following associations:

- A directed with association with one instance of Message

2.7.1.2 Implementations
This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.7.2 Attribute(s)
2.7.2.1 messageRelationshipType (zero or one)
messageRelationshipType is a classification that characterizes the relationship that is established between two Messages.

Valid values:

- A (SX001G:acknowledgementOfMessage)
- O (SX001G:observationOnMessage)
- R (SX001G:replyToMessage)
- U (SX001G:updateToMessage)

Source:

- SX001G

Chapter 7.3.19

Units of functionality - UoF Mission Definition

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.12	Units of functionality - UoF Environment Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.24	Units of functionality - UoF Product Usage Context
Chap 7.3.25	Units of functionality - UoF Product Usage Phase
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.32	Units of functionality - UoF Task
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

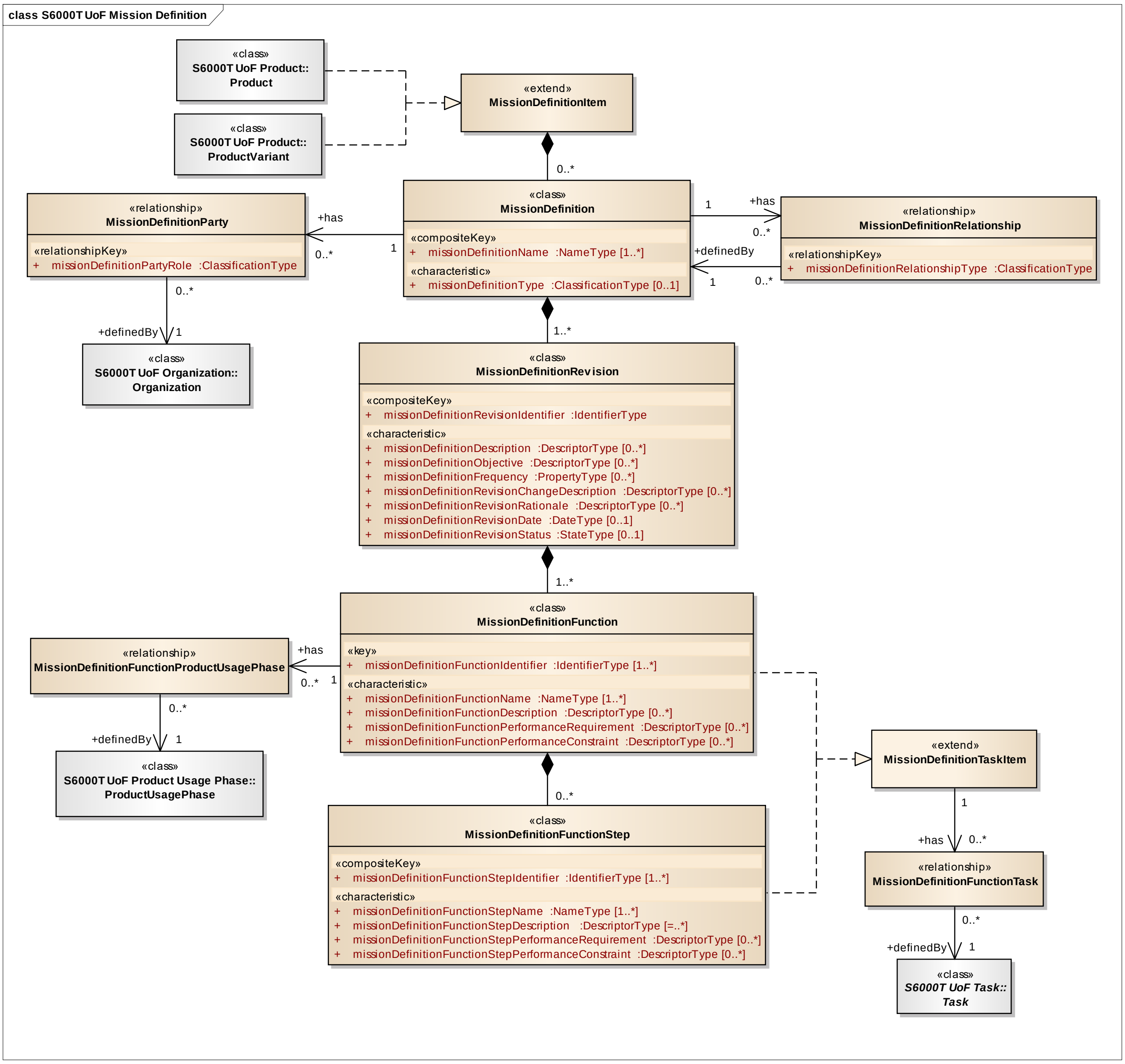
1 General

The Mission Definition UoF supports the definition of operational scenarios to be carried out by a Product together with descriptions on how a person, or body of persons, must interact with the Product to achieve its defined operational scenario objectives.

2 UoF Mission Definition

Key features of the UoF Mission Definition data model, (refer to Fig 1), are:

- define missions which will be carried out or be supported by a Product
- defines situations where one or many persons must interact with the Product during a defined mission
- identify during which Product usage phases these human interactions are required
- detail distinct interactions in the format of one or many tasks



ICN-S6000T-B6865-00039-002-01

Fig 1 UoF Mission Definition

2.1 MissionDefinition

2.1.1 Class definition

MissionDefinition is a <<class>> that represents the defining information for a Product operational scenario.

Examples:

- long haul flight
- search and rescue
- troop transport

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of MissionDefinitionRevision
- A directed has association with zero, one or many instances of MissionDefinitionParty
- A directed has association with zero, one or many instances of MissionDefinitionRelationship

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- EnvironmentDefinitionItem. Refer to Chap 7.3.12.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 missionDefinitionName (one or many)

missionDefinitionName is a name by which the MissionDefinition is known and can be easily referenced.

Source:

- SX001G

2.1.2.2 missionDefinitionType (zero or one)

missionDefinitionType is a classification that identifies further specialization for the MissionDefinition.

Valid values:

- SAR (SX001G:searchAndRescueMission)
- SRV (SX001G:surveillanceMission)

Source:

- SX001G

2.2 MissionDefinitionFunction

2.2.1 Class definition

MissionDefinitionFunction is a <<class>> that represents a specific situation during the mission where a person, or group of persons, must interact with the Product (refer to Chap 7.3.23).

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of MissionDefinitionFunctionStep

- A directed has association with zero, one or many instances of MissionDefinitionFunctionProductUsagePhase

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- MissionDefinitionTaskItem
- PerformanceParameterItem. Refer to Chap 7.3.22.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 missionDefinitionFunctionIdentifier (one or many)

missionDefinitionFunctionIdentifier is an identifier that establishes a unique designator for a MissionDefinitionFunction and to differentiate it from other instances of MissionDefinitionFunction.

Valid identifier class values:

- ID (SX001G:missionDefinitionFunctionIdentifier)

2.2.2.2 missionDefinitionFunctionName (one or many)

missionDefinitionFunctionName is a name by which the MissionDefinitionFunction is known and can be easily referenced.

2.2.2.3 missionDefinitionFunctionDescription (zero, one or many)

missionDefinitionFunctionDescription is a description that provides more information on the MissionDefinitionFunction.

2.2.2.4 missionDefinitionFunctionPerformanceRequirement (zero, one or many)

missionDefinitionFunctionPerformanceRequirement is a description that defines the required behavior, degree, and standard from the person(s) interacting with the Product (refer to Chap 7.3.23).

2.2.2.5 missionDefinitionFunctionPerformanceConstraint (zero, one or many)

missionDefinitionFunctionPerformanceConstraint is a description that defines possible limitations in the performance of the person(s) interacting with the Product (refer to Chap 7.3.23), relative to the performance requirements for the MissionDefinitionFunction.

2.3 MissionDefinitionFunctionProductUsagePhase

2.3.1 Class definition

MissionDefinitionFunctionProductUsagePhase is a <<relationship>> that associates a MissionDefinitionFunction with a ProductUsagePhase during which the MissionDefinitionFunction can or must be carried out.

2.3.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of ProductUsagePhase. Refer to Chap 7.3.25).

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4 MissionDefinitionFunctionStep

2.4.1 Class definition

MissionDefinitionFunctionStep is a <<class>> that represents a specific activity performed as part of the MissionDefinitionFunction.

2.4.1.1 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- MissionDefinitionTaskItem
- PerformanceParameterItem. Refer to Chap 7.3.22.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 missionDefinitionFunctionStepIdentifier

missionDefinitionFunctionStepIdentifier is an identifier that establishes a unique designator for a MissionDefinitionFunctionStep and to differentiate it from other instances of MissionDefinitionFunctionStep.

Valid identifier class values:

- ID (SX001G:missionDefinitionFunctionStepIdentifier)

2.4.2.2 missionDefinitionFunctionStepName (one or many)

missionDefinitionFunctionStepName is a name by which the MissionDefinitionFunctionStep is known and can be easily referenced.

2.4.2.3 missionDefinitionFunctionStepDescription

missionDefinitionFunctionStepDescription is a description that provides more information on the MissionDefinitionFunctionStep.

2.4.2.4 missionDefinitionFunctionStepPerformanceRequirement (zero, one or many)
missionDefinitionFunctionStepPerformanceRequirement is a description that defines the required behavior, degree and standard from the person(s) interacting with the Product (refer to Chap 7.3.23).

2.4.2.5 missionDefinitionFunctionStepPerformanceConstraint (zero, one or many)
missionDefinitionFunctionStepPerformanceConstraint is a description that defines possible limitations in the performance of the person(s) interacting with the Product (refer to Chap 7.3.23), relative to the performance requirements for the MissionDefinitionFunctionStep.

2.5 MissionDefinitionFunctionTask

2.5.1 Class definition

MissionDefinitionFunctionTask is a <<relationship>> that associates the MissionDefinitionFunction with a Task (refer to Chap 7.3.32) that can or must be carried out as part of the MissionDefinitionFunction.

2.5.1.1 Associations
This class has the following associations:

- A directed definedBy association with one instance of Task. Refer to Chap 7.3.32.

2.5.1.2 Implementations
This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.6 MissionDefinitionItem

2.6.1 Interface definition

MissioDefinitionItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.6.1.1 Associations
The MissioDefinitionItem <<extend>> interface has the following associations:

- An aggregate association with zero, one or many instances of MissionDefinition

2.6.1.2 Class members
Classes that implement the MissioDefinitionItem <<extend>> interface are:

- Product. Refer to Chap 7.3.23.
- ProductVariant. Refer to Chap 7.3.23.

2.7 MissionDefinitionParty

2.7.1 Class definition

MissionDefinitionParty is a <<relationship>> that establishes an association between a MissionDefinition and an Organization (refer to Chap 7.3.20).

Source:

– SX001G

2.7.1.1 Associations

This class has the following associations:

– A directed definedBy association with one instance of Organization (refer to Chap 7.3.20)

2.7.1.2 Implementations

This class implements the following <<extend>> interfaces:

– DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
– ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.7.2 Attributes

2.7.2.1 missionDefinitionPartyRole

missionDefinitionPartyRole is a classification that identifies the purpose for MissionDefinitionParty in relation to the MissionDefinition.

Valid values:

– C (SX001G:missionDefinitionCarriedOutBy)

Source:

– SX001G

2.8 MissionDefinitionRelationship

2.8.1 Class definition

MissionDefinitionRelationship is a <<relationship>> where one MissionDefinition relates to another MissionDefinition.

Note

Mission definition relationship can be used to represent the relationship between an overarching mission definition and sub-mission definitions.

Example:

– helicopter search and rescue sub-mission definitions for search, locate and rescue, respectively

Source:

– SX001G

2.8.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of MissionDefinition

2.8.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.8.2 Attributes

2.8.2.1 missionDefinitionRelationshipType

missionDefinitionRelationshipType is a classification that identifies the meaning of the established relationship.

Source:

- SX001G

Valid values:

- DEP (SX001G:dependentOnMissionDefinition)
- EXT (SX001G:extensionOfMissionDefinition)
- SUB (SX001G:subordinateMissionDefinition)

2.9 MissionDefinitionRevision

2.9.1 Class definition

MissionDefinitionRevision is a <<class>> representing an iteration applied to a MissionDefinition.

Source:

- SX001G

2.9.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of MissionDefinitionFunction

2.9.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- EnvironmentDefinitionItem. Refer to Chap 7.3.12.
- PerformanceParameterItem. Refer to Chap 7.3.22.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.9.2 Attributes

2.9.2.1 missionDefinitionRevisionIdentifier

missionDefinitionRevisionIdentifier is an identifier that establishes a unique designator for a MissionDefinitionRevision and to differentiate it from other instances of MissionDefinitionRevision.

Valid identifier class values:

- ID (SX001G:missionDefinitionRevisionIdentifier)

Source:

- SX001G

2.9.2.2 missionDefinitionDescription (zero, one or many)

missionDefinitionDescription is a description of the Product (refer to Chap 7.3.23) operational scenario.

Source:

- SX001G

2.9.2.3 missionDefinitionObjective (zero, one or many)

missionDefinitionObjective is a description that specifies the expected end state after MissionDefinition completion.

Source:

- SX001G

2.9.2.4 missionDefinitionFrequency (zero, one or many)

missionDefinitionFrequency is a property that specifies the rate of occurrence for the MissionDefinition.

Valid unit value:

- timeCycleRateUnit

Source:

- SX001G

2.9.2.5 missionDefinitionRevisionChangeDescription (zero, one or many)

missionDefinitionRevisionChangeDescription is a description that gives more information on content that has been altered between two revisions of MissionDefinition.

2.9.2.6 missionDefinitionRevisionRationale (zero, one or many)

missionDefinitionRevisionRationale is a description that gives more information on the justification for revising the MissionDefinition.

Source:

- SX001G

2.9.2.7 missionDefinitionRevisionDate (zero or one)

missionDefinitionRevisionDate is a date that specifies when a MissionDefinition was revised.

Source:

- SX001G

2.9.2.8 missionDefinitionRevisionStatus (zero or one)
missionDefinitionRevisionStatus is a state that identifies the maturity of a MissionDefinitionRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

Source:

- SX001G

2.10 MissionDefinitionTaskItem

2.10.1 Interface definition

MissionDefinitionTaskItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

2.10.1.1 Associations
The MissionDefinitionTaskItem <<extend>> interface has the following associations:

- A directed definedBy association with one instance of MissionDefinitionFunctionTask

2.10.1.2 Class members
Classes that implement the MissionDefinitionTaskItem <<extend>> interface are:

- MissionDefinitionFunction
- MissionDefinitionFunctionStep

Chapter 7.3.20

Units of functionality - UoF Organization

Table of contents

Page

List of tables

List of figures

References

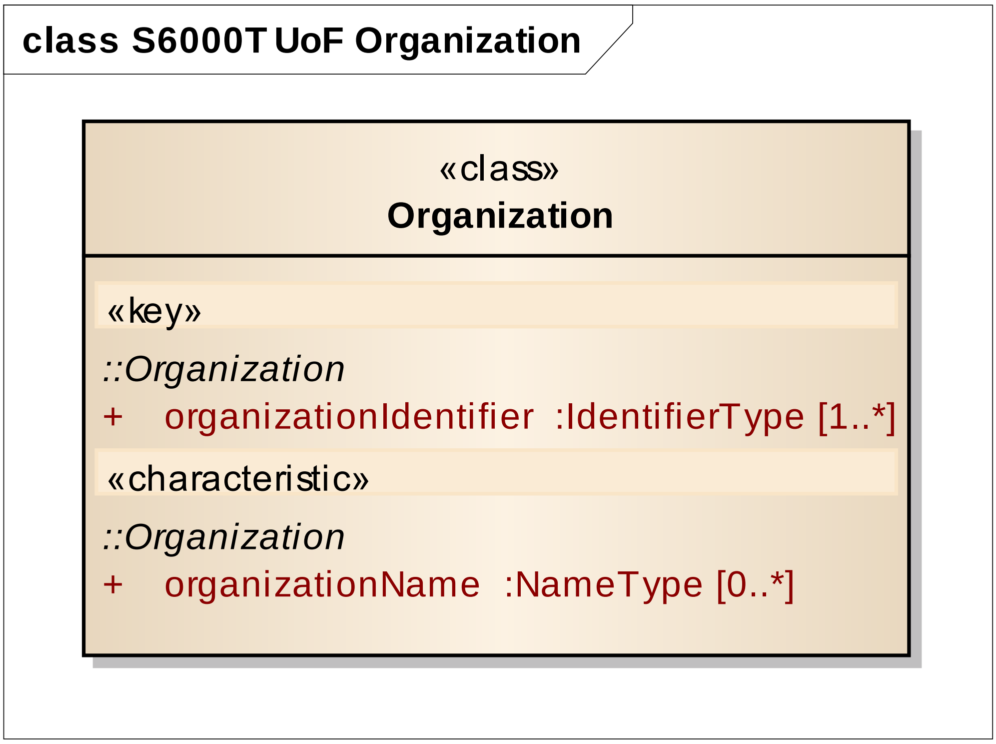
Table 1 References

Chap No./Document No.	Title
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Organization UoF provides the capability to identify organizations.

2 UoF Organization



ICN-S6000T-B6865-00062-001-01

Fig 1 UoF Organization

2.1 Organization

2.1.1 Class definition

`Organization` is a `<<class>>` that represents an administrative structure with a particular purpose belonging to a legal entity.

Examples:

- government department
- international agency
- company
- department

Source:

- SX001G

2.1.1.1 Implementations

This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 organizationIdentifier (one or many)

organizationIdentifier is an identifier that establishes a unique designator for an Organization and allows it to be differentiated from other instances of Organization.

Valid identifier class values:

- CAGE (SX001G:cageCode)
- ID (SX001G:organizationIdentifier)

Source:

- SX001G

2.1.2.2 organizationName (zero, one or many)

organizationName is a name by which the Organization is known and can be easily referenced.

Source:

- SX001G

Chapter 7.3.21

Units of functionality - UoF Part Definition

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark

Chap 7.3.28	Units of functionality - UoF Security Classification
S3000L	International procedure specification for Logistic Support Analysis (LSA)
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

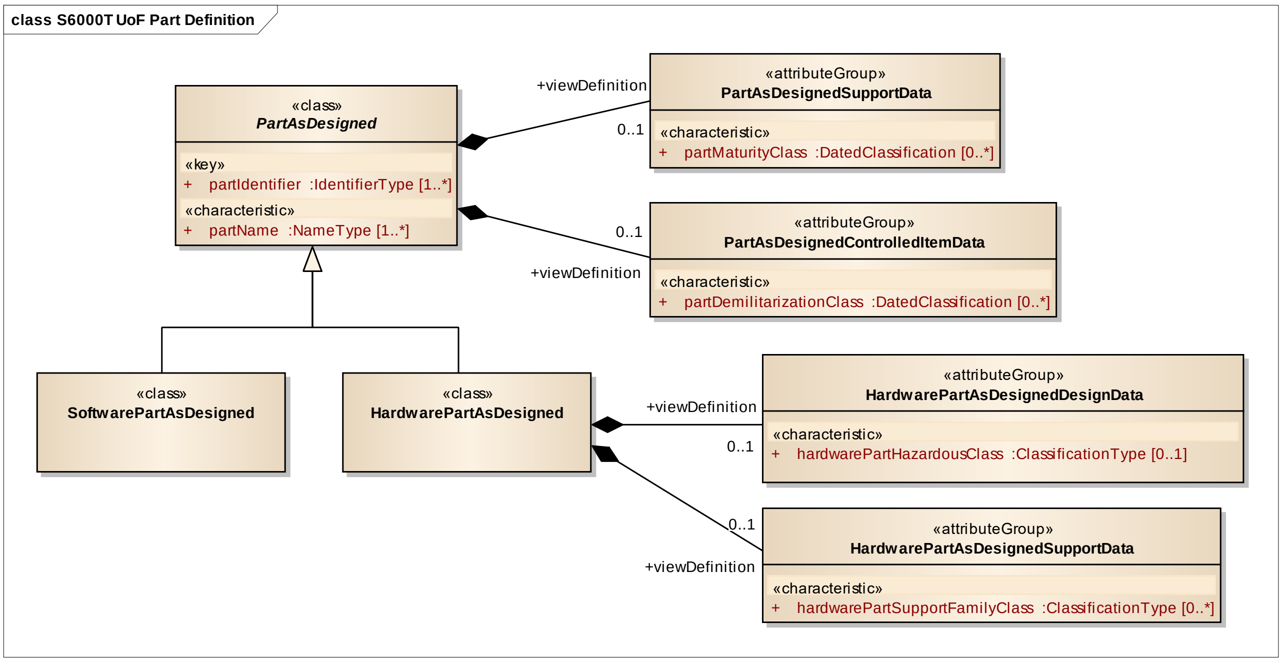
1 General

The Part Definition UoF provides the capability of defining hardware and software parts and their characteristics.

2 UoF Part Definition

Key features of the UoF Part Definition data model, (refer to Fig 1), are:

- a part as designed represents the definitional information for an artifact fulfilling a set of requirements, which can be produced or realized
- a part as designed must be defined as either a hardware part as designed, or a software part as designed



ICN-S6000T-B6865-00034-001-01

Fig 1 UoF Part Definition

2.1 HardwarePartAsDesigned

2.1.1 Class definition

`HardwarePartAsDesigned` is a `PartAsDesigned` that is to be realized as physical items, including non-countable material.

Note

Examples of non-countable materials are, oil, sealant, paint.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate viewDefinition association with zero or one instance of PartAsDesignedControlledItemData (inherited from PartAsDesigned)
- An aggregate viewDefinition association with zero or one instance of PartAsDesignedControlledItemData (inherited from PartAsDesigned)
- An aggregate viewDefinition association with zero or one instance of HardwarePartAsDesignedDesignData
- An aggregate viewDefinition association with zero or one instance of HardwarePartAsDesignedSupportData

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem (inherited from PartAsDesigned). Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from PartAsDesigned). Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 partIdentifier (one or many)

Inherited from PartAsDesigned.

2.1.2.2 partName (one or many)

Inherited from PartAsDesigned.

2.2 HardwarePartAsDesignedDesignData

2.2.1 Attribute group definition

HardwarePartAsDesignedDesignData is an <<attributeGroup>> that collects HardwarePartAsDesigned characteristics identified during design activities.

Source:

- SX001G

2.2.2 Attributes

2.2.2.1 hardwarePartHazardousClass (zero or one)

hardwarePartHazardousClass is a classification that identifies to what extent a HardwarePartAsDesigned is capable of posing a significant risk to health, safety or property during transportation, handing or storage.

Valid values:

- Refer to the S6000T valid values XML schema for details

Source:

- SX001G

2.3 HardwarePartAsDesignedSupportData

2.3.1 Attribute group definition

HardwarePartAsDesignedSupportData is an <<attributeGroup>> that collects HardwarePartAsDesigned characteristics identified during supportability analysis activities.

Source:

- SX001G

2.3.2 Attributes

2.3.2.1 hardwarePartSupportFamilyClass (zero, one or many)

hardwarePartSupportFamilyClass is a classification that generalizes the HardwarePartAsDesigned from a support analysis perspective.

Note

hardwarePartSupportFamilyClass can be used to organize hardware parts that share similar support characteristics or even share the same maintenance solution.

Valid values:

- GBX (SX001G:gearboxPartSupportFamilyClass)
- HTR (SX001G:heaterPartSupportFamilyClass)
- VLV (SX001G:valvePartSupportFamilyClass)

Source:

- S3000L

2.4 PartAsDesigned

2.4.1 Class definition

PartAsDesigned is a <<class>> that represents the definitional information for an artifact fulfilling a set of requirements, which can be produced or realized.

Source:

- SX001G

2.4.1.1 Associations

This class has the following associations:

- An aggregate viewDefinition association with zero or one instance of PartAsDesignedControlledItemData
- An aggregate viewDefinition association with zero or one instance of PartAsDesignedControlledItemData

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

- SecurityClassificationItem. Refer to Chap 7.3.28.

2.4.2 Attributes

2.4.2.1 partIdentifier (one or many)

partIdentifier is an identifier that establishes a unique designator for a PartAsDesigned and to differentiate it from other instances of PartAsDesigned.

Note

Part identification includes drawing, model, type or source controlling numbers.

Example:

- "12345-501"

Valid identifier class values:

- ID (SX001G:partIdentifier)
- OEM (SX001G:originalEquipmentManufacturerPartNumber)
- REF (SX001G:partReferenceNumber)
- STD (SX001G:standardsReferenceDesignator)
- SUP (SX001G:supplierPartNumber)

Source:

- SX001G

2.4.2.2 partName (one or many)

partName is a name by which the PartAsDesigned is known and can be easily referenced.

Source:

- SX001G

2.5 PartAsDesignedControlledItemData

2.5.1 Attribute group definition

PartAsDesignedControlledItemData is an <<attributeGroup>> that collects PartAsDesigned characteristics identified during different controlled item analysis activities.

Source:

- S3000L

2.5.2 Attributes

2.5.2.1 partDemilitarizationClass (zero, one or many)

partDemilitarizationClass is a classification defining special measures to be taken when a PartAsDesigned is being disposed of.

Note

Render them useless for military purposes or destroy any indications of military purposes or performance characteristics.

Valid values:

- A (SX001G:demilitarizationNotRequiredPart)
- B (SX001G:tradeSecurityControlAtDisposalPart)

- C (SX001G:keyPointsDemilitarizationPart)
- D (SX001G:mutilationDemilitarizationPart)
- E (SX001G:nationalFurnishedDemilitarizationPart)
- F (SX001G:managerFurnishedDemilitarizationPart)
- G (SX001G:demilitarizationPriorToTransferPart)
- R (SX001G:specificInstructionsDemilitarizationPart)
- Y (SX001G:specificInstructionsDemilitarizationCryptoMaterial)

Source:

- S3000L

2.6 PartAsDesignedSupportData

2.6.1 Attribute group definition

PartAsDesignedSupportData is an <<attributeGroup>> that collects PartAsDesigned characteristics identified during supportability analysis activities.

Source:

- S3000L

2.6.2 Attributes

2.6.2.1 partMaturityClass (zero, one or many)

partMaturityClass is a classification that defines the maturity of the PartsAsDesigned to determine the certainty by which its characteristics can be valued.

Valid values:

- COTS (SX001G:commonOfTheShelfPart)
- CUST (SX001G:customerFurnishedPart)
- MAM (SX001G:majorModificationOfExistingPart)
- MOM (SX001G:moderateModificationOfExistingPart)
- NEW (SX001G:newDevelopedPart)
- OBS (SX001G:obsoletePart)

Source:

- S3000L

2.7 SoftwarePartAsDesigned

2.7.1 Class definition

SoftwarePartAsDesigned is a PartAsDesigned that is produced as an executable software or as a data file.

Note

Non-executable software includes eg, maps.

Source:

- SX001G

2.7.1.1 Associations

This class has the following associations:

- An aggregate viewDefinition association with zero or one instance of PartAsDesignedControlledItemData (inherited from PartAsDesigned)

– An aggregate `viewDefinition` association with zero or one instance of `PartAsDesignedControlledItemData` (inherited from `PartAsDesigned`)

2.7.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `DigitalFileReferencingItem` (inherited from `PartAsDesigned`). Refer to Chap 7.3.10.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.
- `SecurityClassificationItem` (inherited from `PartAsDesigned`). Refer to Chap 7.3.28.

2.7.2 Attributes

2.7.2.1 partIdentifier (one or many)

Inherited from `PartAsDesigned`.

2.7.2.2 partName (one or many)

Inherited from `PartAsDesigned`.

Chapter 7.3.22

Units of functionality - UoF Performance Parameter

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.32	Units of functionality – UoF Task

Chap No./Document No.	Title
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

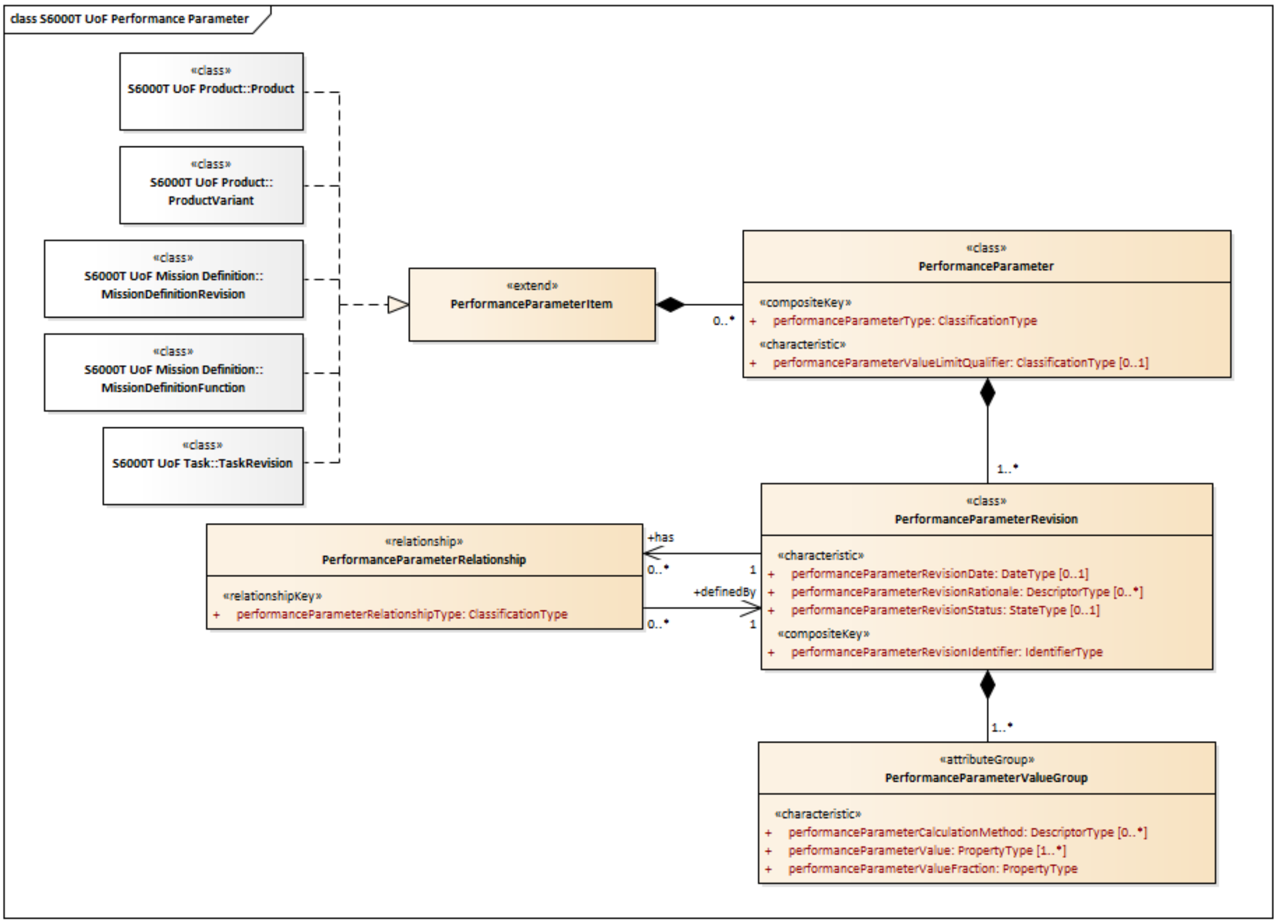
1 General

The Performance Parameter UoF supports the definition of metrics that if changed will have a major impact on the system performance, schedule, cost and/or risk.

2 UoF Performance Parameter

Key features of the UoF Performance Parameter data model, (refer to Fig 1), are:

- define performance parameters associated with the Product and/or its mission definitions
- the definition of a performance parameter can be iterated using `PerformanceParameterRevision`
- an individual performance parameter can have multiple values eg, with different applicability



ICN-S6000T-B6865-00040-003-01

Fig 1 UoF Performance Parameter

2.1 PerformanceParameter

2.1.1 Class definition

PerformanceParameter is a <<class>> that defines a metrics that if changed, or will not be fulfilled, will have a major impact on the system performance, schedule, cost and/or risk.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of PerformanceParameterRevision

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 performanceParameterType

performanceParameterType is a classification that identifies the type of PerformanceParameter being exchanged.

Examples:

- response time
- turnaround time

Valid values:

- TDTT (SX001G:taskDelayToleranceTime)

Source:

- SX001G

2.1.2.2 performanceParameterValueLimitQualifier (zero or one)

performanceParameterValueLimitQualifier is a classification that specifies a directed limit to be applied when testing a value against the defined PerformanceParameter.

Valid values:

- MAX (SX001G:maximumValueLimitQualifier)
- MIN (SX001G:minimumValueLimitQualifier)

Source:

- SX001G

2.2 PerformanceParameterItem

2.2.1 Interface definition

PerformanceParameterItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.2.1.1 Associations

The PerformanceParameterItem <<extend>> interface has the following associations:

- An aggregate association with zero, one or many instances of PerformanceParameter

2.2.1.2 Class members

Classes that implement the PerformanceParameterItem <<extend>> interface are:

- MissionDefinitionFunction. Refer to Chap 7.3.19.
- MissionDefinitionRevision. Refer to Chap 7.3.19.
- Product. Refer to Chap 7.3.23.
- ProductVariant. Refer to Chap 7.3.23.
- TaskRevision. Refer to Chap 7.3.32.

2.3 PerformanceParameterRelationship

2.3.1 Class definition

PerformanceParameterRelationship is a <<relationship>> where one PerformanceParameter relates to another PerformanceParameter.

2.3.1.1 Associations

This class has the following associations:

- A directed association with one instance of PerformanceParameterRevision

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 performanceParameterRelationshipType

performanceParameterRelationshipType is a classification that identifies the meaning of the established relationship.

Valid values:

- BO (SX001G:performanceParameterValueBasedOn)
- DF (SX001G:performancePrameterValueDerivedFrom)

2.4 PerformanceParameterRevision

2.4.1 Class definition

PerformanceParameterRevision is a <<class>> representing an iteration applied to a PerformanceParameter.

Source:

- SX001G

2.4.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of PerformanceParameterValueGroup
- A directed has association with zero, one or many instances of PerformanceParameterRelationship

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 performanceParameterRevisionIdentifier

performanceParameterRevisionIdentifier is an identifier that establishes a unique designator for a PerformanceParameterRevision and to differentiate it from other instances of PerformanceParameterRevision.

Valid identifier class values:

- ID (SX001G:performanceParameterRevisionIdentifier)

Source:

- SX001G

2.4.2.2 performanceParameterRevisionRationale (zero, one or many)

performanceParameterRevisionRationale is a description that gives more information on the justification for revising the defined PerformanceParameter and its values.

Source:

- SX001G

2.4.2.3 performanceParameterRevisionDate (zero or one)

performanceParameterRevisionDate is a date that specifies when a PerformanceParameterRevision was defined.

Source:

- SX001G

2.4.2.4 performanceParameterRevisionStatus (zero or one)

performanceParameterRevisionStatus is a state that identifies the maturity of a PerformanceParameterRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

Source:

- SX001G

2.5 PerformanceParameterValueGroup

2.5.1 Attribute group definition

PerformanceParameterValueGroup is an <<attributeGroup>> that organizes PerformanceParameter values according to a defined purpose.

Source:

- SX001G

2.5.1.1 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.

2.5.2 Attributes

2.5.2.1 performanceParameterValue

performanceParameterValue is a property that represents a value which is determined for the PerformanceParameter.

Valid unit value:

- unit

Source:

- SX001G

2.5.2.2 performanceParameterValueFraction

performanceParameterValueFraction is a property that represents the fraction of all occurrences related to a specified PerformanceParameter that must be within the limit of the defined performanceParameterValue.

Example:

- a customer requirement is that 98% of all replacement tasks must be performed below a specified value of two hours (= maximum replacement time)

Valid unit value:

- relativeUnit

Source:

- SX001G

2.5.2.3 performanceParameterCalculationMethod (zero, one or many)

`performanceParameterCalculationMethod` is a description that gives more information on the method by which the `performanceParameterValue` has been derived.

Source:

- SX001G

Chapter 7.3.23

Units of functionality - UoF Product

Table of contents

Page

List of tables

List of figures

References

Table 1 References

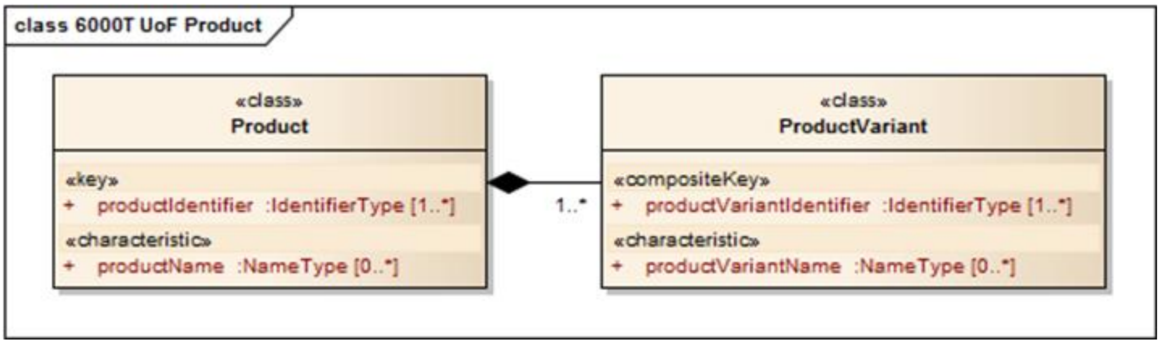
Chap No./Document No.	Title
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.22	Units of functionality - UoF Performance Parameter
Chap 7.3.25	Units of functionality - UoF Product Usage Phase
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Product UoF defines the Product(s) which are in focus for the Integrated Product Support (IPS) program.

2 UoF Product

A key feature of the UoF Product data model, (refer to Fig 1), is that Once defined, a Product will then come in one or many Product variants.



ICN-S6000T-B6865-00027-001-01

Fig 1 UoF Product

2.1 Product

2.1.1 Class definition

`Product` is a <<class>> that represents a family of items which share the same underlying design purpose.

Examples:

- Aegis Class Destroyer
- Airbus A340
- Ford Fusion
- iPhone 12
- Pegasus engine
- Stryker

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of `ProductVariant`

2.1.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `BreakdownItem`. Refer to Chap 7.3.3.
- `DigitalFileReferencingItem`. Refer to Chap 7.3.10.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `MissionDefinitionItem`. Refer to Chap 7.3.19.
- `PerformanceParameterItem`. Refer to Chap 7.3.22.
- `ProductUsagePhaseItem`. Refer to Chap 7.3.25.

- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 productIdentifier (one or many)

productIdentifier is an identifier that establishes a unique designator for a Product and to differentiate it from other instances of Product.

Valid identifier class values:

- EIAC (SX001G:endItemAcronymCode)
- ID (SX001G:productIdentifier)
- MOI (SX001G:modelIdentificationCode)

Source:

- SX001G

2.1.2.2 productName (zero, one or many)

productName is a name by which the Product is known and can be easily referenced.

Source:

- SX001G

2.2 ProductVariant

2.2.1 Class definition

ProductVariant is a <<class>> that defines a member of a Product family which is configured for a specific purpose and is made available to the market.

Note

A product variant is often known as a model.

Examples:

- Boeing 787-800 vs 787-900
- Ford Fusion S vs. SE vs. SEL

Source:

- SX001G

2.2.1.1 Implementations

This class implements the following <<extend>> interfaces:

- BreakdownItem. Refer to Chap 7.3.3.
- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- MissionDefinitionItem. Refer to Chap 7.3.19.
- PerformanceParameterItem. Refer to Chap 7.3.22.
- ProductUsagePhaseItem. Refer to Chap 7.3.25.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

– SecurityClassificationItem. Refer to Chap 7.3.28.

2.2.2 Attributes

2.2.2.1 productVariantIdentifier (one or many)

productVariantIdentifier is an identifier that establishes a unique designator for a ProductVariant and to differentiate it from other instances of ProductVariant.

Source:

– SX001G

Valid identifier class values:

– ID (SX001G:productVariantIdentifier)
– MOI (SX001G:modelIdentificationCode)
– UOC (SX001G:usableOnCode)

2.2.2.2 productVariantName (zero, one or many)

productVariantName is a name by which the ProductVariant is known and can be easily referenced.

Source:

– SX001G

Chapter 7.3.24

Units of functionality - UoF Product Usage Context

Table of contents

Page

List of tables

List of figures

References

Table 1 References

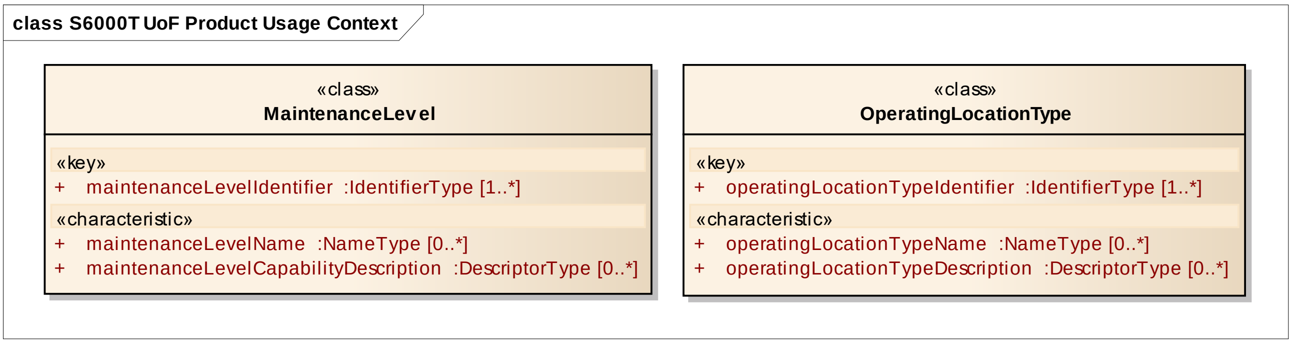
Chap No./Document No.	Title
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Product Usage Context UoF defines the context in which the defined Product(s) and Product variant(s) are to be operated and maintained.

2 UoF Product Usage Context

A key feature of the UoF Product Usage Context data model, (refer to Fig 1), is that locations where a Product can be operated and maintained can be defined in terms of maintenance levels and/or operating location types.



ICN-S6000T-B6865-00056-001-01

Fig 1 UoF Product Usage Context

2.1 MaintenanceLevel

2.1.1 Class definition

MaintenanceLevel is a <<class>> that represents the definition of a set of maintenance capabilities which will be made available to support a defined Product (refer to Chap 7.3.23).

Note

MaintenanceLevel might be established either by a single organization or be distributed between a set of organizations.

Source:

- SX001G

2.1.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 maintenanceLevelIdentifier (one or many)

maintenanceLevelIdentifier is an identifier that establishes a unique designator for a MaintenanceLevel and to differentiate it from other instances of MaintenanceLevel.

Valid identifier class values:

- ID (SX001G:maintenanceLevelIdentifier)

Source:

- SX001G

2.1.2.2 maintenanceLevelName (zero, one or many)

maintenanceLevelName is a name by which the MaintenanceLevel is known and can be easily referenced.

Source:

- SX001G

2.1.2.3 maintenanceLevelCapabilityDescription (zero, one or many)
`maintenanceLevelCapabilityDescription` is a description that gives more information on the ability to perform maintenance based on availability of support resources and environmental conditions.

Note

The defined abilities are the basis in determining the functions to be accomplished at the defined maintenance level.

Note

Support resources includes eg, personnel and skills, special facilities, and support equipment.

Source:

- SX001G

2.2 OperatingLocationType

2.2.1 Class definition

`OperatingLocationType` is a `<<class>>` that represents the definition of the nature of the environment in which a product will be operated.

Source:

- SX001G

2.2.1.1 Implementations
This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 operatingLocationTypeIdentifier (one or many)
`operatingLocationTypeIdentifier` is an identifier that establishes a unique designator for an `OperatingLocationType` and to differentiate it from other instances of `OperatingLocationType`.

Valid identifier class values:

- `ID` (SX001G:operatingLocationTypeIdentifier)

Source:

- SX001G

2.2.2.2 operatingLocationTypeName (zero, one or many)
`operatingLocationTypeName` is a name by which the `OperatingLocationType` is known and can be easily referenced.

Source:

- SX001G

2.2.2.3 operatingLocationTypeDescription (zero, one or many)

`operatingLocationTypeDescription` is a description that gives more information on the `OperatingLocationType`, including environmental conditions to be expected.

Source:

- SX001G

Chapter 7.3.25

Units of functionality - UoF Product Usage Phase

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

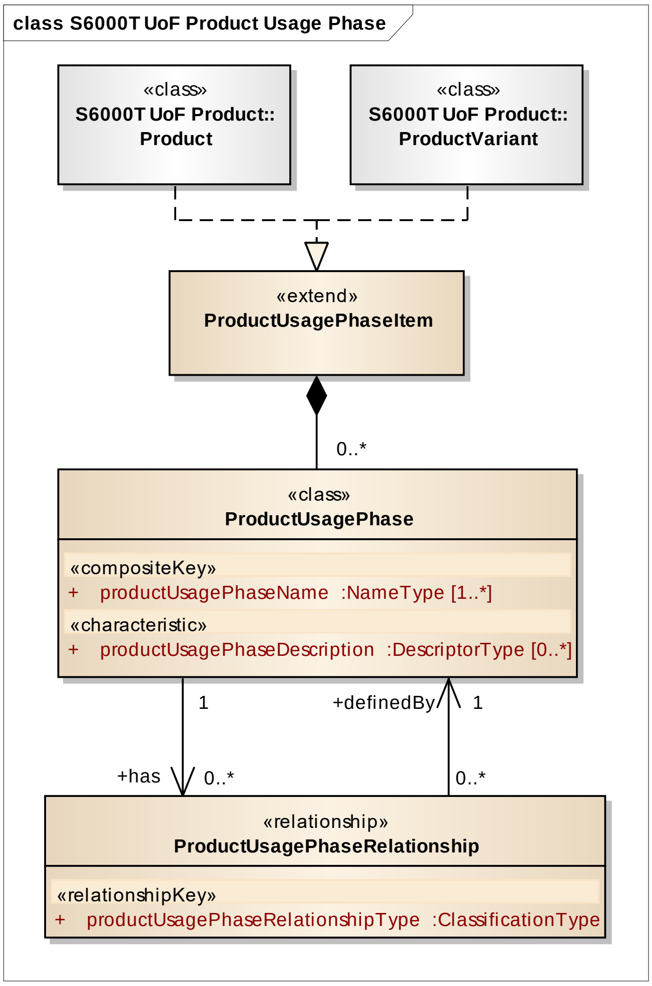
1 General

The Product Usage Phase UoF defines periods of time during which a Product is in an operational state which has specific characteristics that need special considerations.

2 UoF Product Usage Phase

Key features of the UoF Product Usage Phase data model, (refer to Fig 1), are:

- Product usage phases can be defined for the Product or per its Product variants
- Product usage phases can be organized using `ProductUsagePhaseRelationship`



ICN-S6000T-B6865-00028-001-01

Fig 1 UoF Product Usage Phase

2.1 ProductUsagePhase

2.1.1 Class definition

`ProductUsagePhase` is a `<<class>>` that represents a distinct period of time during which a `ProductUsagePhaseItem` is in an operational `state` which has specific characteristics that need special considerations.

Examples:

- emersion, surface, and dock for a submarine
- preflight, postflight, cruise, taxiing for an aircraft

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

– A directed `has` association with zero, one or many instances of `ProductUsagePhaseRelationship`

2.1.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.
- `SecurityClassificationItem`. Refer to Chap 7.3.28.

2.1.2 Attributes

2.1.2.1 productUsagePhaseName (one or many)

`productUsagePhaseName` is a name by which the `ProductUsagePhase` is known and can be easily referenced.

Source:

- SX001G

2.1.2.2 productUsagePhaseDescription (zero, one or many)

`productUsagePhaseDescription` is a description that gives more information on the `ProductUsagePhase`.

Source:

- SX001G

2.2 ProductUsagePhaseItem

2.2.1 Interface definition

`ProductUsagePhaseItem` is an `<<extend>>` interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.2.1.1 Associations

The `ProductUsagePhaseItem` `<<extend>>` interface has the following associations:

- An aggregate association with zero, one or many instances of `ProductUsagePhase`

2.2.1.2 Class members

Classes that implement the `ProductUsagePhaseItem` `<<extend>>` interface are:

- `Product`. Refer to Chap 7.3.23.
- `ProductVariant`. Refer to Chap 7.3.23.

2.3 ProductUsagePhaseRelationship

2.3.1 Class definition

`ProductUsagePhaseRelationship` is a `<<relationship>>` where one `ProductUsagePhase` relates to another `ProductUsagePhase`.

Source:

– SX001G

2.3.1.1 Associations

This class has the following associations:

– A directed `definedBy` association with zero, one or many instances of `ProductUsagePhase`

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

– `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
– `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
– `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.3.2 Attribute

2.3.2.1 productUsagePhaseRelationshipType

`productUsagePhaseRelationshipType` is a classification that identifies the meaning of the established relationship.

Valid values:

– `A` (SX001G:happensAfterProductUsagePhase)
– `B` (SX001G:happensBeforeProductUsagePhase)
– `D` (SX001G:happensDuringProductUsagePhase)
– `P` (SX001G:partOfProductUsagePhase)

Source:

– SX001G

Chapter 7.3.26

Units of functionality - UoF Remark

Table of contents

Page

List of tables

List of figures

References

Table 1 References

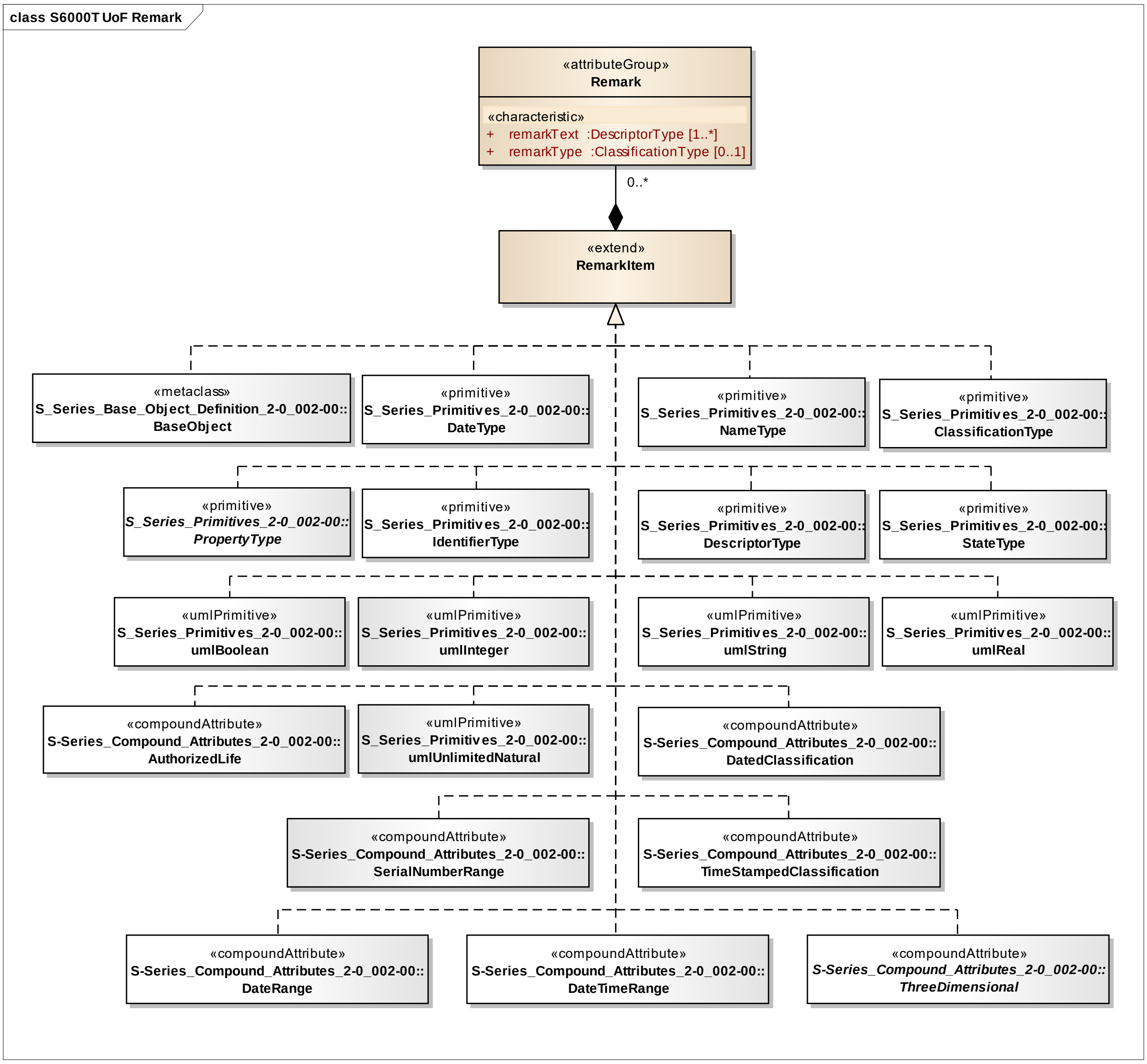
Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.5	Units of functionality - UoF Circuit Breaker
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.12	Units of functionality - UoF Environment Definition
Chap 7.3.13	Units of functionality - UoF Hardware Element
Chap 7.3.14	Units of functionality - UoF Job Duty

Chap No./Document No.	Title
Chap 7.3.15	Units of functionality - UoF Learning Assessment
Chap 7.3.16	Units of functionality - UoF Learning Gap
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.18	Units of functionality - UoF Message
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.20	Units of functionality - UoF Organization
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.22	Units of functionality - UoF Performance Parameter
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.24	Units of functionality - UoF Product Usage Context
Chap 7.3.25	Units of functionality - UoF Product Usage Phase
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.29	Units of functionality - UoF Software Element
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
Chap 7.3.35	Units of functionality - UoF Task Resource
Chap 7.3.36	Units of functionality - UoF Task Target
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization
Chap 7.3.39	Units of functionality - UoF Training Entry Requirement
Chap 7.3.40	Units of functionality - UoF Training Gap
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity
Chap 7.3.42	Units of functionality - UoF Training Media Resource
Chap 7.3.43	Units of functionality - UoF Training Method
Chap 7.3.44	Units of functionality - UoF Training System Resource Requirement
Chap 7.3.45	Units of functionality - UoF Warning Caution Train Prioritization
Chap 7.3.46	Units of functionality - UoF Zone Element
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Remark UoF provides the capability to annotate additional information relevant to the associated item which is not part of the immediate subject.

2 UoF Remark



ICN-S6000T-B6865-00064-001-01

Fig 1 UoF Remark

2.1 Remark

2.1.1 Attribute group definition

`Remark` is an `<<attributeGroup>>` that provides additional information about the associated item.

Note

A remark can be a personal opinion ("I prefer more onions in my soup") or it can be a technical fact ("The manufacturer recommends heating the soup to 45 degrees").

Source:

- SX001G

2.1.1.1 Implementations

This attribute group implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.10.

2.1.2 Attributes

2.1.2.1 remarkText (one or many)

remarkText is a description that provides the text of the additional information.

Source:

- SX001G

2.1.2.2 remarkType (zero or one)

remarkType is a classification that defines the purpose of the Remark.

Examples:

- internal note
- technical fact

Valid values:

- INT SX001G:internalRemark
- PUB SX001G:publicRemark
- RSP SX001G:responseToRemark

Source:

- SX001G

2.2 RemarkItem

2.2.1 Interface definition

RemarkItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.2.1.1 Associations

The RemarkItem <<extend>> interface has the following associations:

- An aggregate association with zero, one or many instances of Remark

2.2.1.2 Class members

Classes that implement the RemarkItem <<extend>> interface are:

- ApplicabilityStatement. Refer to Chap 7.3.2.
- AssociatedCompetencyDefinition. Refer to Chap 7.3.31.
- AssociatedEnvironmentDefinition. Refer to Chap 7.3.12.
- AuthorizedLife. Refer to SX002D.
- Breakdown. Refer to Chap 7.3.3.

- BreakdownElement. Refer to Chap 7.3.3.
- BreakdownElementInZone. Refer to Chap 7.3.46.
- BreakdownElementRevision. Refer to Chap 7.3.3.
- BreakdownElementStructure. Refer to Chap 7.3.3.
- BreakdownElementUsageInBreakdown. Refer to Chap 7.3.3.
- BreakdownElementUsageRelationship. Refer to Chap 7.3.3.
- BreakdownRevision. Refer to Chap 7.3.3.
- BreakdownRevisionRelationship. Refer to Chap 7.3.3.
- ChangeAuthorization. Refer to Chap 7.3.4.
- ChangeNotification. Refer to Chap 7.3.4.
- CircuitBreaker. Refer to Chap 7.3.5.
- CircuitBreakerLocation. Refer to Chap 7.3.5.
- CircuitBreakerSetting. Refer to Chap 7.3.32.
- CircuitBreakerSettings. Refer to Chap 7.3.32.
- ClassificationType. Refer to SX002D.
- CompetencyDefinition. Refer to Chap 7.3.6.
- CompetencyFramework. Refer to Chap 7.3.6.
- ConditionInstance. Refer to Chap 7.3.2.
- ConditionType. Refer to Chap 7.3.2.
- ConditionTypeAssertMember. Refer to Chap 7.3.2.
- CourseDefinition. Refer to Chap 7.3.7.
- CourseDefinitionContext. Refer to Chap 7.3.7.
- CourseDefinitionRelationship. Refer to Chap 7.3.7.
- CourseDefinitionRevision. Refer to Chap 7.3.7.
- CourseIncludedCourseElement. Refer to Chap 7.3.7.
- CourseElementDefinition. Refer to Chap 7.3.8.
- CourseElementDefinitionRelationship. Refer to Chap 7.3.8.
- CourseElementDefinitionRevision. Refer to Chap 7.3.8.
- CourseElementLearningObjective. Refer to Chap 7.3.8.
- CurriculumCourse. Refer to Chap 7.3.9.
- CurriculumDefinition. Refer to Chap 7.3.9.
- CurriculumDefinitionContext. Refer to Chap 7.3.9.
- CurriculumDefinitionRelationship. Refer to Chap 7.3.9.
- CurriculumDefinitionRevision. Refer to Chap 7.3.9.
- DatedClassification. Refer to SX002D.
- DateRange. Refer to SX002D.
- DateTimeRange. Refer to SX002D.
- DateType. Refer to SX002D.
- DescriptorType. Refer to SX002D.
- DigitalFile. Refer to Chap 7.3.10.
- DigitalFileReference. Refer to Chap 7.3.10.
- Document. Refer to Chap 7.3.11.
- DocumentIssue. Refer to Chap 7.3.11.
- DutyDefinition. Refer to Chap 7.3.14.
- DutyDefinitionRelationship. Refer to Chap 7.3.14.
- EnvironmentDefinition. Refer to Chap 7.3.12.
- EnvironmentDefinitionCharacteristic. Refer to Chap 7.3.12.
- EnvironmentDefinitionRelationship. Refer to Chap 7.3.12.

- EnvironmentDefinitionRevision. Refer to Chap 7.3.12.
- EvaluationCriteria. Refer to Chap 7.3.2.
- HardwareElementPartRealization. Refer to Chap 7.3.13.
- IdentifierType. Refer to SX002D.
- JobDefinition. Refer to Chap 7.3.14.
- JobDefinitionRevision. Refer to Chap 7.3.14.
- JobTask. Refer to Chap 7.3.14.
- KnowledgeSkillsAttitudeGapDefinition. Refer to Chap 7.3.16.
- KnowledgeSkillsAttitudeGapDefinitionRevision. Refer to Chap 7.3.16.
- LearningObjective. Refer to Chap 7.3.17.
- LearningObjectiveAdressedKsaGap. Refer to Chap 7.3.17.
- LearningObjectiveAssessmentDefinition. Refer to Chap 7.3.15.
- LearningObjectiveAssessmentTestComponent. Refer to Chap 7.3.15.
- LearningObjectiveKeyLearningPoint. Refer to Chap 7.3.17.
- LearningObjectiveRelationship. Refer to Chap 7.3.17.
- LearningObjectiveRevision. Refer to Chap 7.3.17.
- LearningObjectiveTrainingMedia. Refer to Chap 7.3.41.
- LearningObjectiveTrainingMethod. Refer to Chap 7.3.43.
- LearningObjectiveTrainingMethodIteration. Refer to Chap 7.3.43.
- MaintenanceLevel. Refer to Chap 7.3.24.
- Message. Refer to Chap 7.3.18.
- MessageContext. Refer to Chap 7.3.18.
- MessageParty. Refer to Chap 7.3.18.
- MessageRelationship. Refer to Chap 7.3.18.
- MissionDefinition. Refer to Chap 7.3.19.
- MissionDefinitionFunction. Refer to Chap 7.3.19.
- MissionDefinitionFunctionProductUsagePhase. Refer to Chap 7.3.19.
- MissionDefinitionFunctionStep. Refer to Chap 7.3.19.
- MissionDefinitionFunctionTask. Refer to Chap 7.3.19.
- MissionDefinitionParty. Refer to Chap 7.3.19.
- MissionDefinitionRelationship. Refer to Chap 7.3.19.
- MissionDefinitionRevision. Refer to Chap 7.3.19.
- NameType. Refer to SX002D.
- OccupationalBackground. Refer to Chap 7.3.31.
- OccupationalBackgroundKSASubject. Refer to Chap 7.3.31.
- OccupationalBackgroundRevision. Refer to Chap 7.3.31.
- OperatingLocationType. Refer to Chap 7.3.24.
- Organization. Refer to Chap 7.3.20.
- PartAsDesigned. Refer to Chap 7.3.21.
- PerformanceParameter. Refer to Chap 7.3.22.
- PerformanceParameterRelationship. Refer to Chap 7.3.22.
- PerformanceParameterRevision. Refer to Chap 7.3.22.
- Product. Refer to Chap 7.3.23.
- ProductUsagePhase. Refer to Chap 7.3.25.
- ProductUsagePhaseRelationship. Refer to Chap 7.3.25.
- ProductVariant. Refer to Chap 7.3.23.
- PropertyType. Refer to SX002D.

- ReferencedDigitalFile. Refer to Chap 7.3.10.
- ReferencedDocument. Refer to Chap 7.3.11.
- ResourceRealization. Refer to Chap 7.3.27.
- ResourceSpecification. Refer to Chap 7.3.27.
- ResourceSpecificationRevision. Refer to Chap 7.3.27.
- ResultingDataModule. Refer to Chap 7.3.32.
- SecurityClass. Refer to Chap 7.3.28.
- SecurityClassification. Refer to Chap 7.3.28.
- SerialNumberRange. Refer to SX002D.
- SoftwareElementPartRealization. Refer to Chap 7.3.29.
- StateType. Refer to SX002D.
- Subtask. Refer to Chap 7.3.32.
- SubtaskAcceptanceParameter. Refer to Chap 7.3.32.
- SubtaskInZone. Refer to Chap 7.3.32.
- SubtaskJobDefinition. Refer to Chap 7.3.30.
- SubtaskSourceDocument. Refer to Chap 7.3.32.
- SubtaskTarget. Refer to Chap 7.3.32.
- SubtaskTimeline. Refer to Chap 7.3.32.
- SubtaskTrainDecision. Refer to Chap 7.3.30.
- SubtaskTrainStep. Refer to Chap 7.3.30.
- SubtaskWarningCautionNote. Refer to Chap 7.3.32.
- Task. Refer to Chap 7.3.32.
- TaskAttitudeLevelOfLearning. Refer to Chap 7.3.33.
- TaskKnowledgeLevelOfLearning. Refer to Chap 7.3.33.
- TaskKnowledgeSkillsAttitudeGap. Refer to Chap 7.3.16.
- TaskResource. Refer to Chap 7.3.35.
- TaskResourceRelationship. Refer to Chap 7.3.35.
- TaskRevision. Refer to Chap 7.3.32.
- TaskSkillLevelOfLearning. Refer to Chap 7.3.33.
- TaskTarget. Refer to Chap 7.3.36.
- TaskTrainDecision. Refer to Chap 7.3.37.
- TaskRevisionWarningCautionNote. Refer to Chap 7.3.32.
- ThreeDimensional. Refer to SX002D.
- TimeStampedClassification. Refer to SX002D.
- TrainingEntryCompetencyDefinition. Refer to Chap 7.3.39.
- TrainingEntryRequirement. Refer to Chap 7.3.39.
- TrainingGap. Refer to Chap 7.3.40.
- TrainingMediaResource. Refer to Chap 7.3.42.
- TrainingSystemResourceRequirement. Refer to Chap 7.3.44.
- umlBoolean. Refer to SX002D.
- umlInteger. Refer to SX002D.
- umlReal. Refer to SX002D.
- umlString. Refer to SX002D.
- umlUnlimitedNatural. Refer to SX002D.
- WarningCautionNote. Refer to Chap 7.3.32.
- WarningCautionTrainDecision. Refer to Chap 7.3.45.
- WarningCautionTrainJob. Refer to Chap 7.3.45.

Chapter 7.3.27

Units of functionality - UoF Resource Specification

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

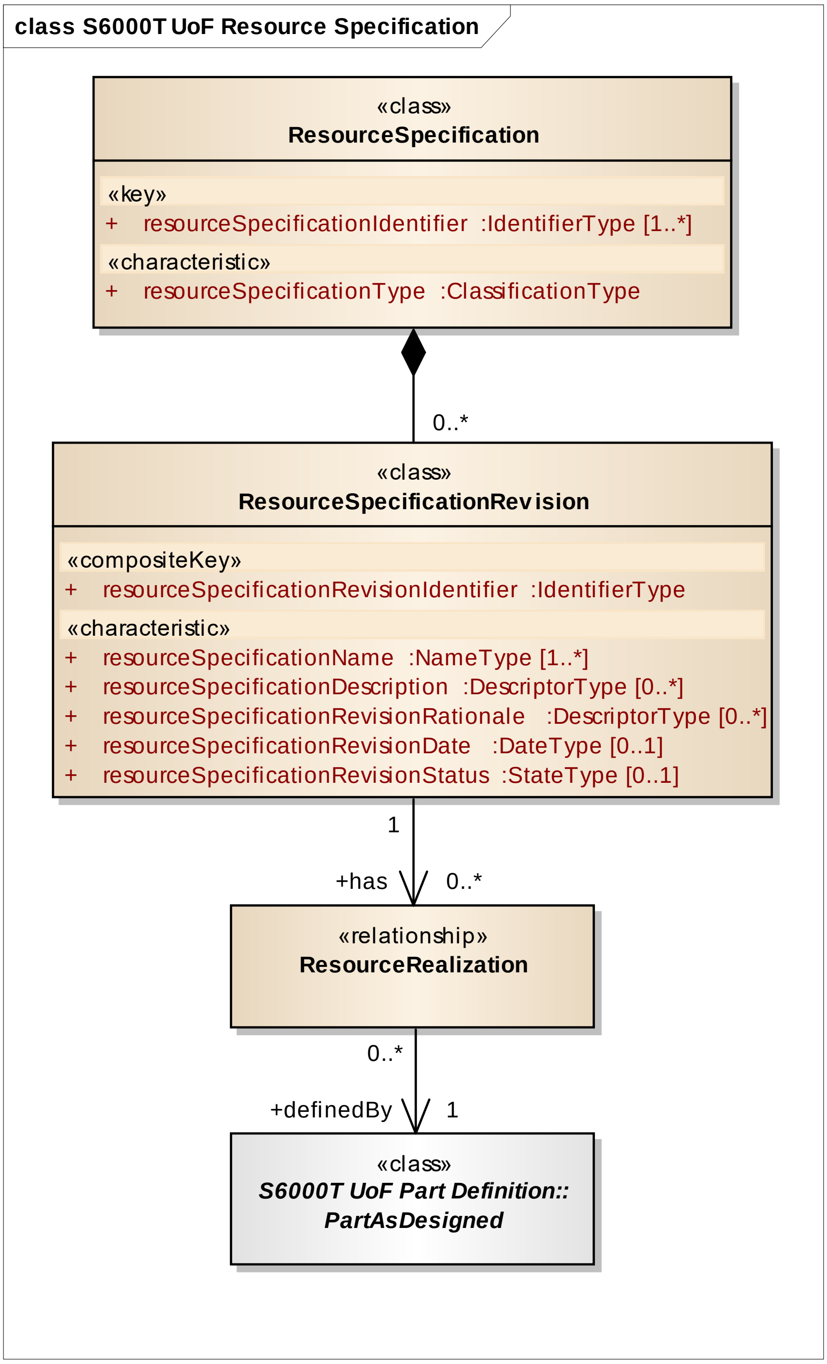
1 General

The Resource Specification UoF provides the capability to define generic definitions of resources needed without having to define its actual realization.

2 UoF Resource Specification

Key features of the UoF Resource Specification data model, (refer to Fig 1), are:

- a `ResourceSpecification` can be iterated (updated) using `ResourceSpecificationRevision`
- an instance of `ResourceSpecificationRevision` can be associated with one or many instances of `PartAsDesigned` that meet the requirements of the specified resource



ICN-S6000T-B6865-00096-001-01

Fig 1 UoF Resource Specification

2.1 ResourceRealization

2.1.1 Class definition

ResourceRealization is a <<relationship>> where a ResourceSpecification relates to an instance of PartAsDesigned that fulfills the resource specification.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of PartAsDesigned

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2 ResourceSpecification

2.2.1 Class definition

ResourceSpecification is a <<class>> that defines a resource by its characteristics.

Note

ResourceSpecification allows for more generic resource definitions ie, a task/subtask does not need to be changed due to eg, customer specific resource preferences.

Source:

- SX001G

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of ResourceSpecificationRevision

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

– SecurityClassificationItem. Refer to Chap 7.3.28.

2.2.2 Attributes

2.2.2.1 resourceSpecificationIdentifier (one or many)

resourceSpecificationIdentifier is an identifier that establishes a unique designator for a ResourceSpecification and to differentiate it from other instances of ResourceSpecification.

Valid identifier class values:

- ID (SX001G:resourceSpecificationIdentifier)
- STD (SX001G:standardsReferenceDesignator)

Source:

- SX001G

2.2.2.2 resourceSpecificationType

resourceSpecificationType is a classification that identifies further specialization for a ResourceSpecification.

Valid values:

- C (SX001G:consumablePart)
- CNWK (SX001G:communicationNetworkInfrastructureResource)
- COMP (SX001G:computerInfrastructureResource)
- DOCK (SX001G:dockInfrastructureResource)
- DRYD (SX001G:dryDockInfrastructureResource)
- E (SX001G:expendableSparePart)
- EP (SX001G:expendablePersonalProtectionPart)
- GAR (SX001G:garageInfrastructureResource)
- HNG (SX001G:hangarInfrastructureResource)
- HT (SX001G:standardHandTool)
- INF (SX001G:infrastructureResource)
- IT (SX001G:informationTechnologyPart)
- M (SX001G:rawMaterial)
- PA (SX001G:packagingPart)
- PE (SX001G:personalProtectionPart)
- POW (SX001G:powerInfrastructureResource)
- PR (SX001G:productProtectionPart)
- R (SX001G:repairableSparePart)
- S (SX001G:sparePart)
- SE (SX001G:supportEquipment)
- SSE (SX001G:safetyRelatedSupportEquipment)
- TNWK (SX001G:transportNetworkInfrastructureResource)

Source:

- SX001G

2.3 ResourceSpecificationRevision

2.3.1 Class definition

ResourceSpecificationRevision is a <<class>> representing an iteration applied to a ResourceSpecification.

Source:

- SX001G

2.3.1.1 Associations

This class has the following associations:

- A directed definedBy association with zero, one or many instances of ResourceRealization

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 resourceSpecificationRevisionIdentifier

resourceSpecificationRevisionIdentifier is an identifier that establishes a unique designator for a ResourceSpecificationRevision and to differentiate it from other instances of ResourceSpecificationRevision.

Valid identifier class values:

- ID (SX001G:resourceSpecificationRevisionIdentifier)

Source:

- SX001G

2.3.2.2 resourceSpecificationName (one or many)

resourceSpecificationName is a name by which the ResourceSpecification is known and can be easily referenced.

Source:

- SX001G

2.3.2.3 resourceSpecificationDescription (zero, one or many)

resourceSpecificationDescription is a description that gives more information on the characteristics that a part realization must fulfill in order to qualify as a possible resource.

Source:

- SX001G

2.3.2.4 resourceSpecificationRevisionRationale (zero, one or many)

`resourceSpecificationRevisionRationale` is a description that gives more information on the justification for revising the `ResourceSpecification`.

Source:

– SX001G

2.3.2.5 resourceSpecificationRevisionDate (zero or one)

`resourceSpecificationRevisionDate` is a date that specifies when the `ResourceSpecification` was revised.

Source:

– SX001G

2.3.2.6 resourceSpecificationRevisionStatus (zero or one)

`resourceSpecificationRevisionStatus` is a state that identifies the maturity of a `ResourceSpecificationRevision`.

Valid state values:

- `A` (SX001G:approvedStatus)
- `D` (SX001G:draftStatus)
- `IW` (SX001G:inWorkStatus)
- `C` (SX001G:cancelledStatus)
- `R` (SX001G:reviewedStatus)

Source:

– SX001G

Chapter 7.3.28

Units of functionality - UoF Security Classification

Table of contents

List of tables

List of figures

References

Table 1 References

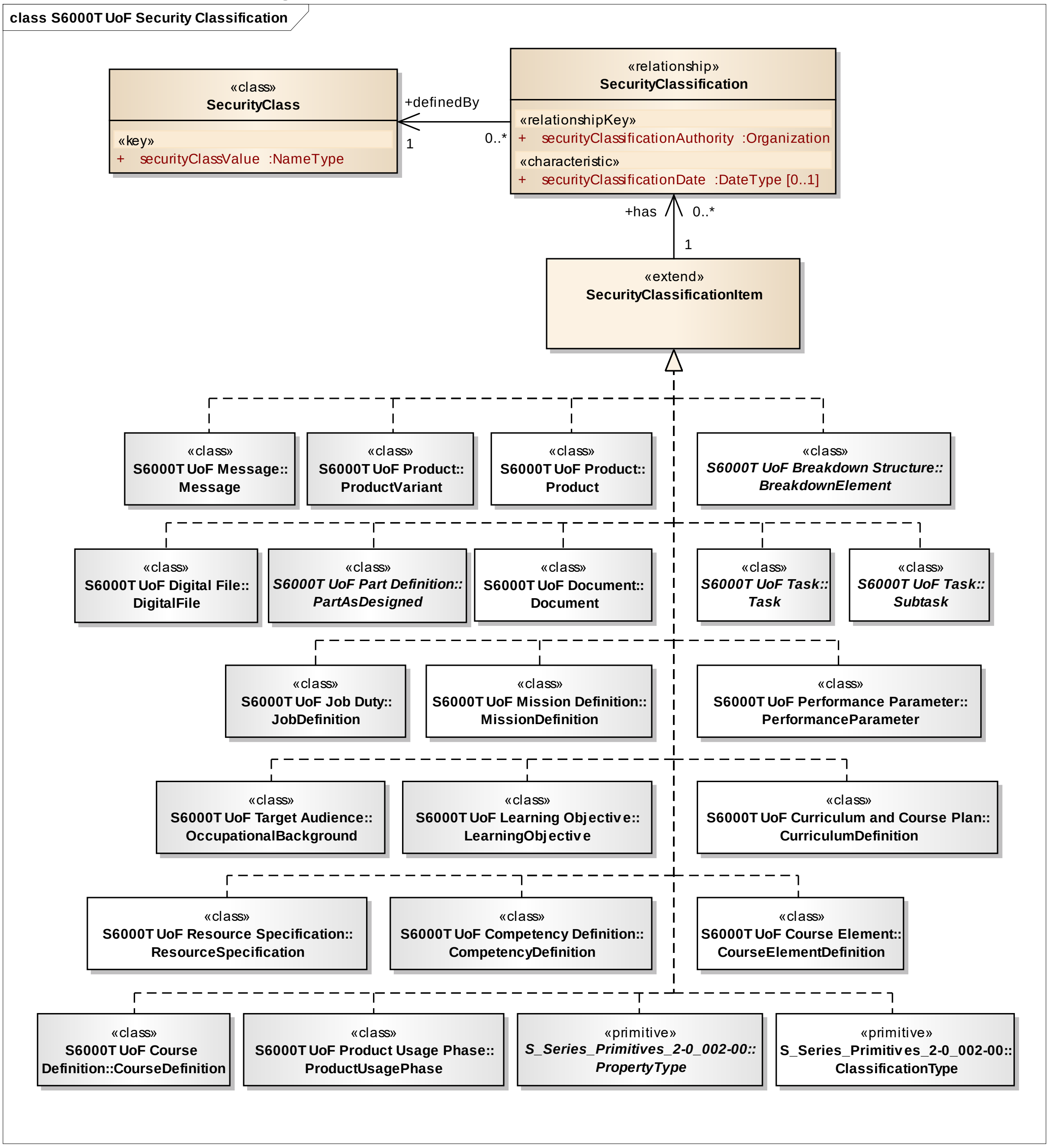
Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.18	Units of functionality - UoF Message

Chap No./Document No.	Title
Chap 7.3.19	Units of functionality - UoF Mission Definition
Chap 7.3.20	Units of functionality - UoF Organization
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.22	Units of functionality - UoF Performance Parameter
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.25	Units of functionality - UoF Product Usage Phase
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.31	Units of functionality - UoF Target Audience
Chap 7.3.32	Units of functionality - UoF Task
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Security Classification UoF provides the capability to assign security classifications to objects that need special handling for protection against unauthorized access or distribution.

2 UoF Security Classification



ICN-S6000T-B6865-00063-002-01

Fig 1 UoF Security Classification

2.1 SecurityClass

2.1.1 Class definition

SecurityClass is a <<class>> that identifies a level of confidentiality which can be used to protect something against unauthorized access.

Source:

– SX001G

2.1.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 securityClassValue

securityClassValue is a name that defines the level of confidentiality.

Examples:

- company confidential
- confidential
- restricted
- secret
- top secret
- unclassified

Source:

- SX001G

2.2 SecurityClassification

2.2.1 Class definition

SecurityClassification is a <<relationship>> that associates a given SecurityClass with the item that must be protected against unauthorized access or distribution

Source:

- SX001G

2.2.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of SecurityClass

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 securityClassificationAuthority

securityClassificationAuthority identifies the Organization (refer to Chap 7.3.20) that is the authoritative source for the defined SecurityClassification.

Source:

- SX001G

2.2.2.2 securityClassificationDate (zero or one)
`securityClassificationDate` is a date when the security classification is declared.

Source:

- SX001G

2.3 SecurityClassificationItem

2.3.1 Interface definition

`SecurityClassificationItem` is an `<<extend>>` interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.3.1.1 Associations
The `SecurityClassificationItem` `<<extend>>` interface has the following associations:

- A directed association with zero, one or many instances of `SecurityClassification`

2.3.1.2 Class members
Classes that implement the `SecurityClassificationItem` `<<extend>>` interface are:

- `BreakdownElement`. Refer to Chap 7.3.3.
- `ClassificationType`. Refer to SX002D.
- `CompetencyDefinition`. Refer to Chap 7.3.6.
- `CourseDefinition`. Refer to Chap 7.3.7.
- `CourseElementDefinition`. Refer to Chap 7.3.8.
- `CurriculumDefinition`. Refer to Chap 7.3.9.
- `DigitalFile`. Refer to Chap 7.3.10.
- `Document`. Refer to Chap 7.3.11.
- `JobDefinition`. Refer to Chap 7.3.14.
- `LearningObjective`. Refer to Chap 7.3.17.
- `Message`. Refer to Chap 7.3.18.
- `MissionDefinition`. Refer to Chap 7.3.19.
- `OccupationalBackground`. Refer to Chap 7.3.31.
- `PartAsDesigned`. Refer to Chap 7.3.21.
- `PerformanceParameter`. Refer to Chap 7.3.22.
- `Product`. Refer to Chap 7.3.23.
- `ProductUsagePhase`. Refer to Chap 7.3.25.
- `ProductVariant`. Refer to Chap 7.3.23.
- `PropertyType`. Refer to SX002D.
- `ResourceSpecification`. Refer to Chap 7.3.27.
- `Subtask`. Refer to Chap 7.3.32.
- `Task`. Refer to Chap 7.3.32.

Chapter 7.3.29

Units of functionality - UoF Software Element

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.46	Units of functionality - UoF Zone Element
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

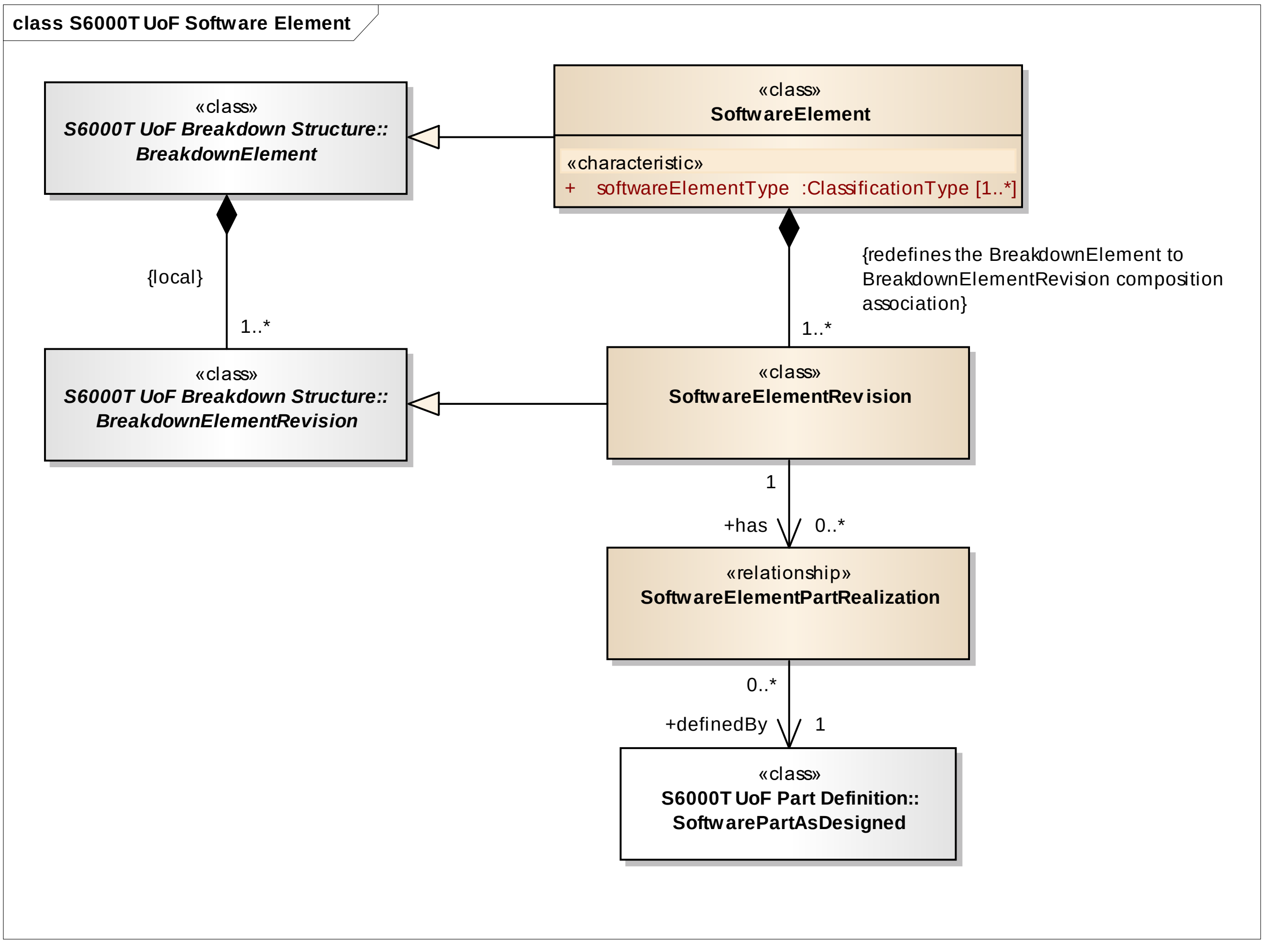
1 General

The Software Element UoF provides the capability to specify that an element within a breakdown is software and can be associated with the software part(s) that fulfill the requirement.

2 UoF Software Element

Key features of the UoF Software Element data model, (refer to Fig 1), are:

- `SoftwareElement` is a specialization of `BreakdownElement` which means that, wherever `BreakdownElement` is being used in the data model, `SoftwareElement` can be used instead
- `SoftwareElementRevision` is a specialization of `BreakdownElementRevision` which means that, wherever `BreakdownElementRevision` is being used in the data model, `SoftwareElementRevision` can be used instead
- an instance of `SoftwareElementRevision` can be associated with one or many instances of `SoftwarePartAsDesigned` that meet the requirements and specification for the `SoftwareElement`



ICN-S6000T-B6865-00031-002-01

Fig 1 UoF Software Element

2.1 SoftwareElement

2.1.1 Class definition

SoftwareElement is a BreakdownElement (refer to Chap 7.3.3) that is realized as a SoftwarePartAsDesigned.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of SoftwareElementRevision

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem (inherited from BreakdownElement). Refer to Chap 7.3.45.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from BreakdownElement). Refer to Chap 7.3.28.

2.2 Attributes

2.2.1.1 breakdownElementIdentifier (one or many)

Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.2.1.2 breakdownElementName (zero, one or many)

Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.2.1.3 breakdownElementEssentiality (zero or many)

Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.2.1.4 softwareElementType (one or many)

softwareElementType is a classification that identifies further specialization for a SoftwareElement.

Valid values:

- D (SX001G:distributedSoftwareElement)
- E (SX001G:embeddedSoftwareElement)
- L (SX001G:loadableSoftwareElement)

Source:

- SX001G

2.3 SoftwareElementPartRealization

2.3.1 Class definition

SoftwareElementPartRealization is a <<relationship>> where a SoftwareElementRevision relates to an instance of SoftwarePartAsDesigned (refer to Chap 7.3.21) which fulfills the SoftwareElement specification.

Source:

- SX001G

2.3.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of SoftwarePartAsDesigned. Refer to Chap 7.3.21.

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4 SoftwareElementRevision

2.4.1 Class definition

SoftwareElementRevision is a BreakdownElementRevision (refer to Chap 7.3.3) representing an iteration applied to a SoftwareElement.

Source:

- SX001G

2.4.1.1 Associations

This class has the following associations:

- A directed has association with zero, one or many instances of SoftwareElementPartRealization

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem (inherited from BreakdownElementRevision). Refer to Chap 7.3.46.
- ChangeControlledItem (inherited from BreakdownElementRevision). Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.

– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 breakdownElementRevisionIdentifier
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.4.2.2 breakdownElementDescription (zero, one or many)
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.4.2.3 breakdownElementRevisionRationale (zero, one or many)
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.4.2.4 breakdownElementRevisionDate (zero or one)
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.4.2.5 breakdownElementRevisionStatus (zero or one)
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

Chapter 7.3.30

Units of functionality - UoF Subtask Train Prioritization

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

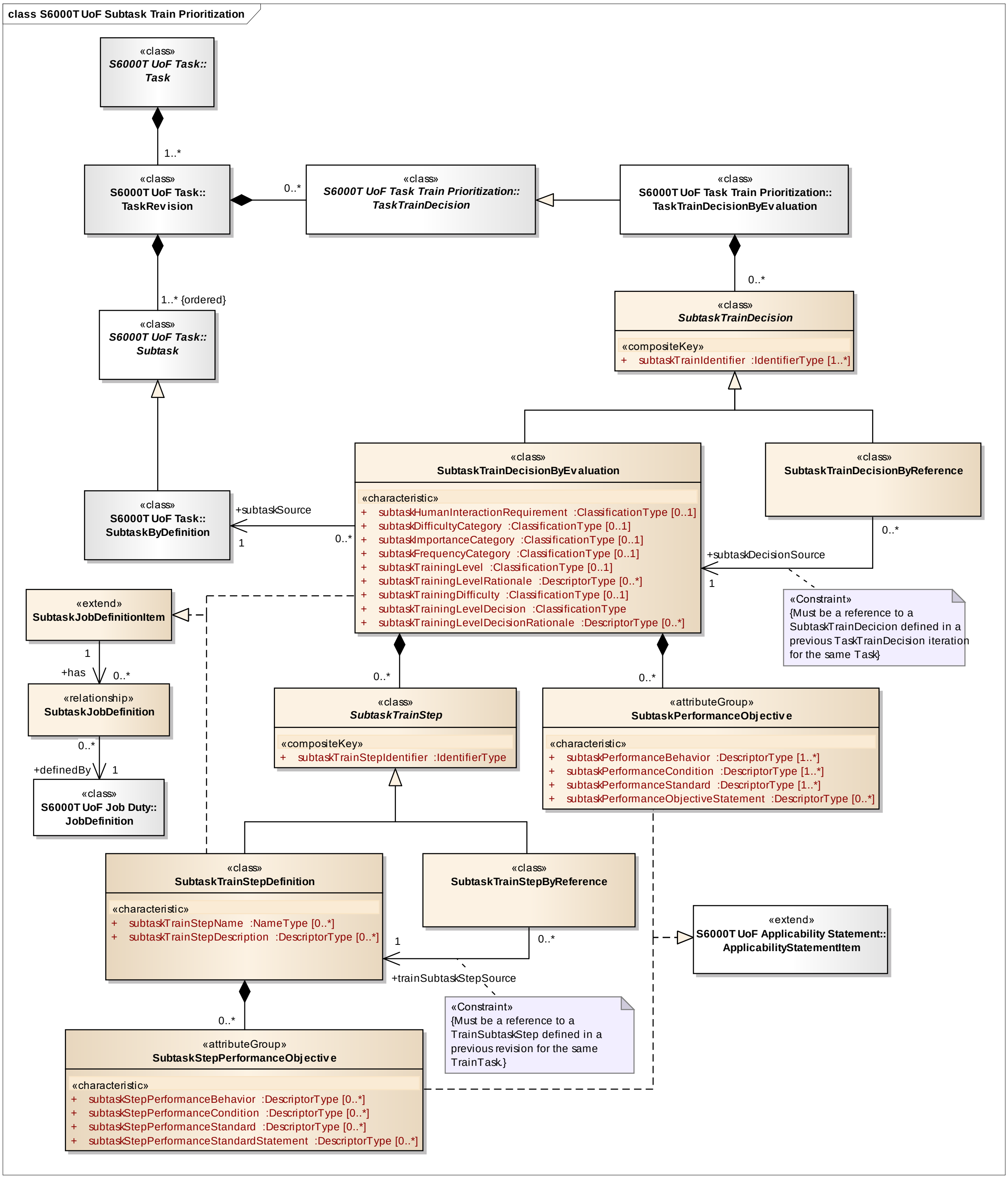
1 General

The Subtask Train Prioritization UoF provides the capability to define training prioritization for a subtask and its subtask steps.

2 UoF Subtask Train Prioritization

Key features of the UoF Subtask Train Prioritization data model, (refer to Fig 1), are:

- Subtask train prioritization can refer to a subtask in a previous task revision for its train prioritization
- Subtask train prioritization can be evaluated using the task difficulty, importance and frequency evaluation model
- Subtasks can be further detailed into subtask steps which need to be evaluated based on its required performance standard



ICN-S6000T-B6865-00044-002-01

Fig 1 UoF Subtask Train Prioritization

2.1 SubtaskJobDefinition

2.1.1 Class definition

SubtaskJobDefinition is a <<relationship>> that associates a SubtaskJobDefinitionItem with a JobDefinition (refer to Chap 7.3.14) which is affected by the train prioritization decision.

2.1.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of JobDefinition

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2 SubtaskJobDefinitionItem

2.2.1 Interface definition

SubtaskJobDefinitionItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

2.2.1.1 Associations

The SubtaskJobDefinitionItem <<extend>> interface has the following associations:

- A directed has association with zero, one or many instances of SubtaskJobDefinition

2.2.1.2 Class members

Classes that implement the SubtaskJobDefinitionItem <<extend>> interface are:

- SubtaskTrainDecisionByEvaluation
- SubtaskTrainStepDefinition

2.3 SubtaskPerformanceObjective

2.3.1 Attribute group definition

SubtaskPerformanceObjective is an <<attributeGroup>> that defines the precise statement of the performance expected in the execution of a Subtask expressed in terms of the subtask performance conditions, behavior and standard.

2.3.1.1 Implementations

This attribute group implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.

2.3.2 Attributes

2.3.2.1 subtaskPerformanceBehavior (one or many)

subtaskPerformanceBehavior is a description that expresses the main purpose for the Subtask.

2.3.2.2 subtaskPerformanceCondition (one or many)
subtaskPerformanceCondition is a description that expresses the environmental, physical and psychological circumstances under which the Subtask is to be performed.

2.3.2.3 subtaskPerformanceStandard (one or many)
subtaskPerformanceStandard is a description that expresses the expected quality and/or time constraints that must be met in the performance of the Subtask.

2.3.2.4 subtaskPerformanceObjectiveStatement (zero, one or many)
subtaskPerformanceObjectiveStatement is a description that expresses the subtaskPerformanceBehavior, subtaskPerformanceCondition and the subtaskPerformanceStandard in one sentence.

2.4 SubtaskStepPerformanceObjective

2.4.1 Attribute group definition

SubtaskStepPerformanceObjective is an <<attributeGroup>> that defines the precise statement of the performance expected in the execution of an action within a Subtask expressed in terms of the subtask step performance conditions, behavior and standard.

2.4.1.1 Implementations
This attribute group implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.

2.4.2 Attributes

2.4.2.1 subtaskStepPerformanceBehavior (zero, one or many)
subtaskStepPerformanceBehavior is a description that identifies the main purpose for the specific step within a Subtask.

2.4.2.2 subtaskStepPerformanceCondition (zero, one or many)
subtaskStepPerformanceCondition is a description that expresses the environmental, physical and psychological circumstances under which a step within a Subtask is to be performed.

2.4.2.3 subtaskStepPerformanceStandard (zero, one or many)
subtaskStepPerformanceStandard is a description that expresses the expected quality and/or time constraints that must be met in the performance of a Subtask step.

2.4.2.4 subtaskStepPerformanceStandardStatement (zero, one or many)
subtaskStepPerformanceStandardStatement is a description that expresses the subtaskStepPerformanceBehavior, subtaskStepPerformanceCondition, and subtaskPerformanceStandard in one sentence.

2.5 SubtaskTrainDecision

2.5.1 Class definition

SubtaskTrainDecision is a <<class>> that defines how a Subtask within a Task is to be managed from a training perspective.

2.5.1.1 Implementations
This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 subtaskTrainIdentifier (one or many)

subtaskTrainIdentifier is an identifier that establishes a unique designator for a SubtaskTrainDecision and to differentiate it from other instances of SubtaskTrainDecision.

Valid identifier class values:

- ID (SX001G:subtaskTrainIdentifier)

2.6 SubtaskTrainDecisionByEvaluation

2.6.1 Class definition

SubtaskTrainDecisionByEvaluation is a SubtaskTrainDecision where the characteristics of the associated Subtask are evaluated in order to decide if the Subtask is subject for training.

2.6.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of SubtaskPerformanceObjective
- An aggregate association with zero, one or many instances of SubtaskTrainStep
- A directed subtaskSource association with zero, one or many instances of SubtaskByDefinition

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to S6000T UoF Document, Chap 7.3.11.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- KnowledgeSkillAttitudeRequirementItem. Refer to Chap 7.3.33.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SubtaskJobDefinitionItem

2.6.2 Attributes

2.6.2.1 subtaskTrainIdentifier (one or many)

Inherited from SubtaskTrainDecision.

2.6.2.2 subtaskHumanInteractionRequirement (zero or one)

subtaskHumanInteractionRequirement is a classification that identifies the type of human coordination that needs to take place when performing the Subtask.

Examples:

- collective
- individual

Valid values:

- `C` SX001G:collectiveSubtask
- `I` SX001G:individualSubtask

2.6.2.3 subtaskDifficultyCategory (zero or one)

`subtaskDifficultyCategory` is a classification that identifies the complexity of a `Subtask` from a human performance perspective.

Note

Subtask difficulty category can be derived from eg, the following subtask aspects:

- Personal safety
- Time it takes to perform the subtask
- Need for human interaction
- Conditions under which the subtask is to be performed

Valid values:

- `H` SX001G:highSubtaskDifficulty
- `L` SX001G:lowSubtaskDifficulty
- `M` SX001G:mediumSubtaskDifficulty

2.6.2.4 subtaskImportanceCategory (zero or one)

`subtaskImportanceCategory` is a classification that identifies possible adverse effects that the performance of the `Subtask` can have with respect to cost (damage) and availability for the item under analysis.

Note

Subtask importance category can be derived from eg, the following task aspects:

- Mission (operability) impact
- Product integrity impact
- Function essentiality

Valid values:

- `H` SX001G:highImportanceSubtask
- `L` SX001G:lowImportanceSubtask
- `M` SX001G:mediumImportanceSubtask

2.6.2.5 subtaskFrequencyCategory (zero or one)

`subtaskFrequencyCategory` is a classification that identifies how often a `Subtask` is to be performed.

Note

Subtask frequency category is based on how many times a subtask is to be performed under a given time period (usually defined per year).

Valid values:

- `I` SX001G:infrequentSubtaskFrequency
- `M` SX001G:moderatlyFrequentSubtask

- V SX001G:veryFrequentSubtask

2.6.2.6 subtaskTrainingLevel (zero or one)
subtaskTrainingLevel is a classification that identifies the priority for training in order for the performer to be able to carry out the Subtask within its defined performance requirement.

Note
Training level is often referred to as training priority.

Valid values:

- 1 SX001G:veryHighPriorityTrainingSubtask
- 2 SX001G:highPriorityTrainingSubtask
- 3 SX001G:moderatePriorityTrainingSubtask
- 4 SX001G:lowPriorityTrainingSubtask
- 5 SX001G:noFormalTrainingRequiredSubtask

2.6.2.7 subtaskTrainingLevelRationale (zero, one or many)
subtaskTrainingLevelRationale is a description that gives more information on the justification for selecting the defined subtask training level.

2.6.2.8 subtaskTrainingDifficulty (zero or one)
subtaskTrainingDifficulty is a classification that identifies the complexity involved in training for the Subtask.

Valid values:

- H SX001G:highDifficultyToTrainSubtask
- L SX001G:lowDifficultyToTrainSubtask
- M SX001G:moderateDifficultyToTrainSubtask

2.6.2.9 subtaskTrainingLevelDecision
subtaskTrainingLevelDecision is a classification that identifies the final decision on subtaskTrainingLevel based on the subtask difficulty, importance and frequency analysis as well as additional considerations such as subtaskTrainingDifficulty, customer stipulations and regulations.

Valid values:

- 1 SX001G:veryHighPriorityTrainingSubtask
- 2 SX001G:highPriorityTrainingSubtask
- 3 SX001G:moderatePriorityTrainingSubtask
- 4 SX001G:lowPriorityTrainingSubtask
- 5 SX001G:noFormalTrainingRequiredSubtask

2.6.2.10 subtaskTrainingLevelDecisionRationale (zero, one or many)
subtaskTrainingLevelDecisionRationale is a description that gives more information on the justification for the final subtaskTrainingLevelDecision.

2.7 SubtaskTrainDecisionByReference

2.7.1 Class definition

SubtaskTrainDecisionByReference is a SubtaskTrainDecision that refers to a previous revision of the Task and Subtask as its source for the subtask training decision.

2.7.1.1 Associations

This class has the following associations:

– A directed subtaskDecisionSourceassociation with one instance of SubtaskTrainDecisionByEvaluation

2.7.1.2 Implementations

This class implements the following <<extend>> interfaces:

– DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
– ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.7.2 Attributes

2.7.2.1 subtaskTrainIdentifier (one or many)

Inherited from SubtaskTrainDecision

2.8 SubtaskTrainStep

2.8.1 Class definition

SubtaskTrainStep is a <<class>> that specifies an action within a Subtask that must be addressed from training perspective.

2.8.1.1 Implementations

This class implements the following <<extend>> interfaces:

– ApplicabilityStatementItem. Refer to Chap 7.3.2.
– DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
– ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.8.2 Attributes

2.8.2.1 subtaskTrainStepIdentifier

subtaskTrainStepIdentifier is an identifier that establishes a unique designator for a SubtaskTrainStep and to differentiate it from other instances of SubtaskTrainStep.

Valid identifier class values:

ID (SX001G:subtaskTrainStepIdentifier)

2.9 SubtaskTrainStepByReference

2.9.1 Class definition

SubtaskTrainStepByReference is a SubtaskTrainStep that refers to a previous revision of the Task and SubtaskTrainStepDefinition as its source for the subtask step training decision.

2.9.1.1 Associations

- A directed trainSubtaskStepSource association with one instance of SubtaskTrainStepDefinition

2.9.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.9.2 Attributes

2.9.2.1 subtaskTrainStepIdentifier

Inherited from SubtaskTrainStep.

2.10 SubtaskTrainStepDefinition

2.10.1 Class definition

SubtaskTrainStepDefinition is a SubtaskTrainStep where the characteristics of a defined step (action) within a Subtask are described in order to decide if the step (action) is to be trained.

2.10.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of SubtaskStepPerformanceObjective

2.10.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- KnowledgeSkillAttitudeRequirementItem. Refer to Chap 7.3.33.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SubtaskJobDefinitionItem

2.10.2 Attributes

2.10.2.1 subtaskTrainStepIdentifier

Inherited from SubtaskTrainStep.

2.10.2.2 subtaskTrainStepName (zero, one or many)

`subtaskTrainStepName` is a name by which the `SubtaskTrainStep` is known and can be easily referenced.

2.10.2.3 subtaskTrainStepDescription (zero, one or many)

`subtaskTrainStepDescription` is a description of a specific action performed as part of a `Subtask`.

Chapter 7.3.31

Units of functionality - UoF Target Audience

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.16	Units of functionality - UoF Learning Gap

Chap No./Document No.	Title
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

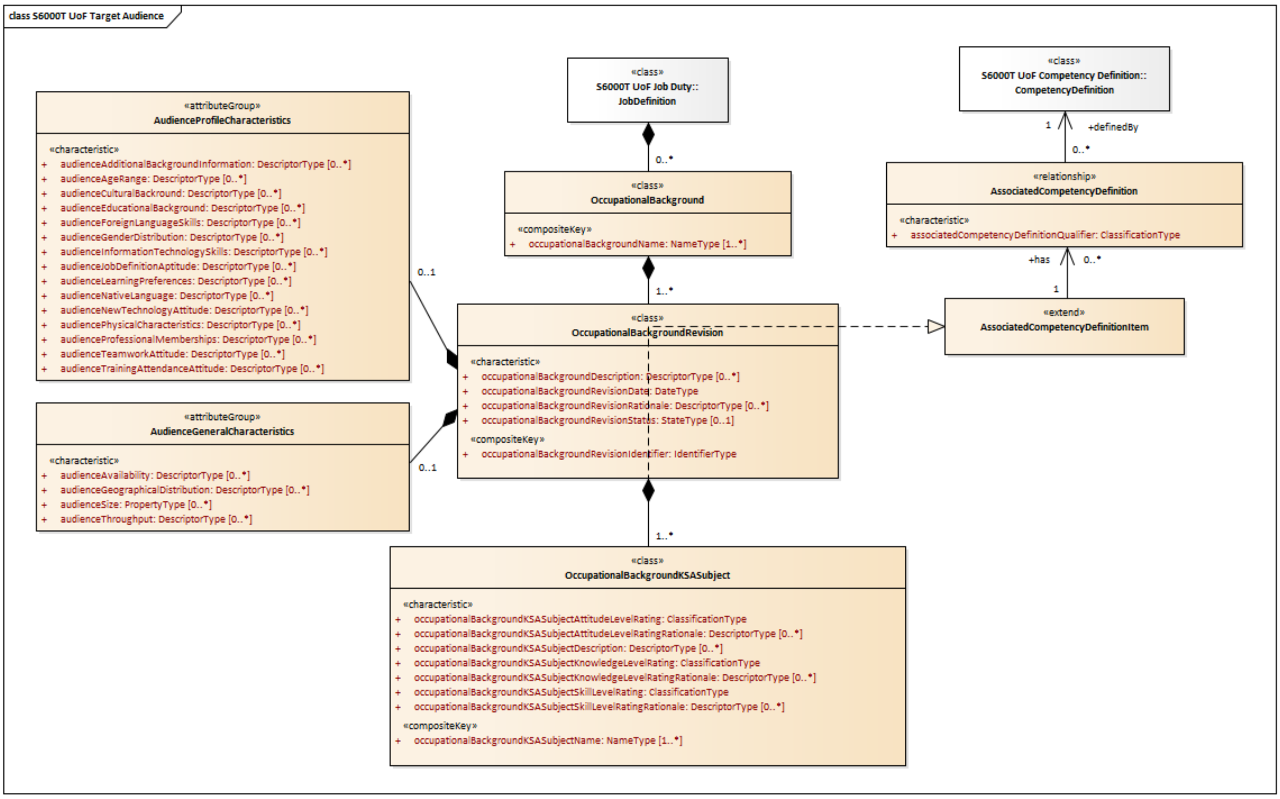
1 General

The Target Audience UoF provides the capability to document existing audience profiles including their existing qualifications and their current knowledge, skills, and attitudes.

2 UoF Target Audience

Key features of the UoF Target Audience data model, (refer to Fig 1), are:

- a job definition can be analyzed from the perspective that persons coming into the training can have different occupational backgrounds and audience profiles
- the description of an occupational background can be further subdivided into a set of subjects where each subject can be associated with the expected KSA ratings for the persons coming into the training
- each occupational background description can be associated with one or many competency definitions eg, expected credentials



ICN-S6000T-B6865-00047-003-01

Fig 1 UoF Target Audience

2.1 AssociatedCompetencyDefinition

2.1.1 Class definition

AssociatedCompetencyDefinition is a <<relationship>> where an instance of OccupationalBackground relates to a defined CompetencyDefinition that is an important part of the description of the OccupationalBackground.

2.1.1.1 Associations

This class has the following associations:

- A directed definedBy association with zero, one or many instances of CompetencyDefinition

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 associatedCompetencyDefinitionQualifier

associatedCompetencyDefinitionQualifier is a classification that determines the level of confidence by which the defined CompetencyDefinition is associated with the OccupationalBackground.

Examples:

- anticipated (Expected)
- identified (Verified)

Valid values:

- A (SX001G:anticipatedCompetency)
- V (SX001G:verifiedCompetency)

2.2 AudienceGeneralCharacteristics

2.2.1 Attribute group definition

AudienceGeneralCharacteristics is an <<attributeGroup>> that collects target population extrinsic characteristics.

2.2.2 Attributes

2.2.2.1 audienceSize (zero, one or many)

audienceSize is a property that specifies the number of persons within the target audience.

Valid unit value:

- countUnit

2.2.2.2 audienceThroughput (zero, one or many)
`audienceThroughput` is a description that provides information about the number of persons within the target audience that will complete a training curriculum withing a specified amount of time.

2.2.2.3 audienceAvailability (zero, one or many)
`audienceAvailability` is a description that provides information about when the target audience for is available for training.

2.2.2.4 audienceGeographicalDistribution (zero, one or many)
`audienceGeographicalDistribution` is a description that provides information about the physical location of the persons within the target audience.

2.3 AudienceProfileCharacteristics

2.3.1 Attribute group definition

`AudienceProfileCharacteristics` is an `<<attributeGroup>>` that collects target population intrinsic characteristics.

2.3.2 Attributes

2.3.2.1 audienceAgeRange (zero, one or many)
`audienceAgeRange` is description that provides information about the lower and upper age boundaries of the persons within the target audience.

2.3.2.2 audienceGenderDistribution (zero, one or many)
`audienceGenderDistribution` is description that provides information about the number of males and females within the target audience.

2.3.2.3 audienceEducationalBackground (zero, one or many)
`audienceEducationalBackground` is a description that provides information about the schools, areas of study, degrees or certificates that are common to the persons within the target audience.

2.3.2.4 audienceTrainingAttendanceAttitude (zero, one or many)
`audienceTrainingAttendanceAttitude` is a description that provides information about the emotions, beliefs, and behaviors of the persons within the target audience in regard to participating in formal training activities.

2.3.2.5 audienceCulturalBackround (zero, one or many)
`audienceCulturalBackround` is a description that provides information about the values, conventions, and social practices that are associated with the environment and surrounding in which the target audience was raised and/or lives within.

2.3.2.6 audiencePhysicalCharacteristics (zero, one or many)
`audiencePhysicalCharacteristics` is a description that provides information about the bodily strength, stamina, coordination, balance, and flexibility of the persons within the target audience.

2.3.2.7 audienceTeamworkAttitude (zero, one or many)
`audienceTeamworkAttitude` is a description that provides information about the emotions, beliefs, and behaviors of the persons within the target audience about working with others versus individually.

2.3.2.8 audienceProfessionalMemberships (zero, one or many)
`audienceProfessionalMemberships` is a description that provides information about the organizations or affiliations to which persons within the target audience belong for a particular job or profession.

2.3.2.9 audienceJobDefinitionAptitude (zero, one or many)
`audienceJobDefinitionAptitude` is a description that provides information about natural talents or special abilities of the people within a target audience in respect to the associated `JobDefinition`.

2.3.2.10 audienceNativeLanguage (zero, one or many)
`audienceNativeLanguage` is a description that provides information about the primary language of the persons within the target audience.

2.3.2.11 audienceForeignLanguageSkills (zero, one or many)
`audienceForeignLanguageSkills` is a description that provides information about the fluency in languages of the persons within the target audience that is in addition to `audienceNativeLanguage`.

2.3.2.12 audienceInformationTechnologySkills (zero, one or many)
`audienceInformationTechnologySkills` is a description that provides information about the ability of the persons within the target audience to use digital systems, equipment, and software.

2.3.2.13 audienceNewTechnologyAttitude (zero, one or many)
`audienceNewTechnologyAttitude` is a description that provides information about the emotions, beliefs, and behaviors of the persons within the target audience in regard to embracing digital systems, equipment, and software that is new to them.

2.3.2.14 audienceLearningPreferences (zero, one or many)
`audienceLearningPreferences` is a description that provides information about the preferred or optimal means through which persons within the target audience learn new knowledge, skills, and attitudes.

2.3.2.15 audienceAdditionalBackgroundInformation (zero, one or many)
`audienceAdditionalBackgroundInformation` is a description that provides other details about the experiences or circumstances of the target audience that is in addition to information described in `audienceCulturalBackround`, `audienceEducationalBackground`, and other `AudienceProfileCharacteristics`.

2.4 OccupationalBackground

2.4.1 Class definition

`OccupationalBackground` is a `<<class>>` that defines perceived knowledge, skills and attitudes common to a set of persons within the target audience designated to perform a certain job.

Note

An occupational background can typically be defined based previous position eg, level, posting or rank.

2.4.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `OccupationalBackgroundRevision`

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- `CurriculumDefinitionTargetAudienceItem`. Refer to Chap 7.3.9.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.
- `SecurityClassificationItem`. Refer to Chap 7.3.28.

2.4.2 Attributes

2.4.2.1 occupationalBackgroundName (one or many)

`occupationalBackgroundName` is a name by which the `OccupationalBackground` is known and can be easily referenced.

2.5 OccupationalBackgroundKSASubject

2.5.1 Class definition

`OccupationalBackgroundKSASubject` is a <<class>> that defines the perceived knowledge, skills and attitudes in a specific field which is important to evaluate and highlight in the context of the defined `JobDefinition`.

2.5.1.1 Associations

This class has the following associations:

- An aggregate association, one or many, to one related object of type `KnowledgeSkillsAttitudeGapDefinition` (refer to Chap 7.3.16)

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- `AssociatedCompetencyDefinitionItem`. Refer to Chap 7.3.6.
- `DigitalFileReferencingItem`. Refer to Chap 7.3.10.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 occupationalBackgroundKSASubjectName (one or many)

`occupationalBackgroundKSASubjectName` is a name by which the `OccupationalBackgroundKSASubject` is known and can be easily referenced.

2.5.2.2 occupationalBackgroundKSASubjectDescription (zero, one or many)

`occupationalBackgroundKSASubjectDescription` is a description that provides further information on a specific field of experience.

2.5.2.3 occupationalBackgroundKSASubjectKnowledgeLevelRating

`occupationalBackgroundKSASubjectKnowledgeLevelRating` is a classification of a person's expected level of sophistication with respect to mental, cognitive, and/or logical aspects in the context of the defined field of experience.

Valid values:

– Refer to the S6000T valid values XML schema for details

2.5.2.4 occupationalBackgroundKSASubjectKnowledgeLevelRatingRationale (zero, one or many)

`occupationalBackgroundKSASubjectKnowledgeLevelRatingRationale` is a description that gives more information on the justification for assigning the knowledge level rating for the defined `OccupationalBackgroundKSASubject`.

2.5.2.5 occupationalBackgroundKSASubjectSkillLevelRating

`occupationalBackgroundKSASubjectSkillLevelRating` is a classification of a person's expected level of sophistication with respect to physical, mechanical, and/or movement-based aspects in the context of the defined field of experience.

Valid values:

– Refer to the S6000T valid values XML schema for details

2.5.2.6 occupationalBackgroundKSASubjectSkillLevelRatingRationale (zero, one or many)

`occupationalBackgroundKSASubjectSkillLevelRatingRationale` is a description that gives more information on the justification for assigning the skill level rating for the defined `OccupationalBackgroundKSASubject`.

2.5.2.7 occupationalBackgroundKSASubjectAttitudeLevelRating

`occupationalBackgroundKSASubjectAttitudeLevelRating` is a classification of a person's expected level of sophistication with respect to emotional, motivational and/or social abilities in the context of the defined field of experience.

2.5.2.8 occupationalBackgroundKSASubjectAttitudeLevelRatingRationale (zero, one or many)

`occupationalBackgroundKSASubjectAttitudeLevelRatingRationale` is a description that gives more information on the justification for assigning the attitude level rating for the defined `OccupationalBackgroundKSASubject`.

2.6 OccupationalBackgroundRevision

2.6.1 Class definition

`OccupationalBackgroundRevision` is a `<<class>>` representing an iteration applied to the definition of an `OccupationalBackground`.

2.6.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of `AudienceGeneralCharacteristics`
- An aggregate association with zero or one instance of `AudienceProfileCharacteristics`
- An aggregate association with one or many instances of `OccupationalBackgroundKSASubject`
- A directed `has` association with zero, one or many instances `AssociatedCompetencyDefinition`

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 occupationalBackgroundRevisionIdentifier

occupationalBackgroundRevisionIdentifier is an identifier that establishes a unique designator for an OccupationalBackgroundRevision and to differentiate it from other instances of OccupationalBackgroundRevision.

Valid identifier class values:

- ID (SX001G:occupationalBackgroundRevisionIdentifier)

2.6.2.2 occupationalBackgroundDescription (zero, one or many)

occupationalBackgroundDescription is a description that provides further information on the perceived knowledge, skills and attitudes common to a set of persons within the target population designated to perform a certain job.

2.6.2.3 occupationalBackgroundRevisionRationale (zero, one or many)

occupationalBackgroundRevisionRationale is a description that gives more information on the justification for revising the defined OccupationalBackground.

2.6.2.4 occupationalBackgroundRevisionDate

occupationalBackgroundRevisionDate is a date that specifies when the OccupationalBackground was revised.

2.6.2.5 occupationalBackgroundRevisionStatus (zero or one)

occupationalBackgroundRevisionStatus is a state that identifies the maturity of the OccupationalBackgroundRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

2.7 AssociatedCompetencyDefinitionItem

2.7.1 Interface definition

AssociatedCompetencyDefinition is an <<extend>> that provides its associated data model to those classes that can be associated with a CompetencyDefinition.

2.7.1.1 Associations

The AssociatedCompetencyDefinitionItem <<extend>> interface has the following associations:

– A directed has association with zero, one or many instances of AssociatedCompetencyDefinition

2.7.1.2 Class members
Classes that implement the AssociatedCompetencyDefinitionItem <<extend>> interface are:

– OccupationalBackgroundKSASubject

Chapter 7.3.32

Units of functionality - UoF Task

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.5	Units of functionality - UoF Circuit Breaker
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.12	Units of functionality - UoF Environment Definition
Chap 7.3.13	Units of functionality - UoF Hardware Element
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.35	Units of functionality - UoF Task Resource
Chap 7.3.45	Units of functionality - UoF Warning Caution Train Prioritization

Chap No./Document No.	Title
S1000D	International specification for technical publications using a common source database
S3000L	International procedure specification for Logistics Support Analysis (LSA)
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Task UoF supports the detailed definition of tasks required to support and operate the Product.

Note

Tasks required to support and operate the Product also include tasks which are specifically developed for training purposes.

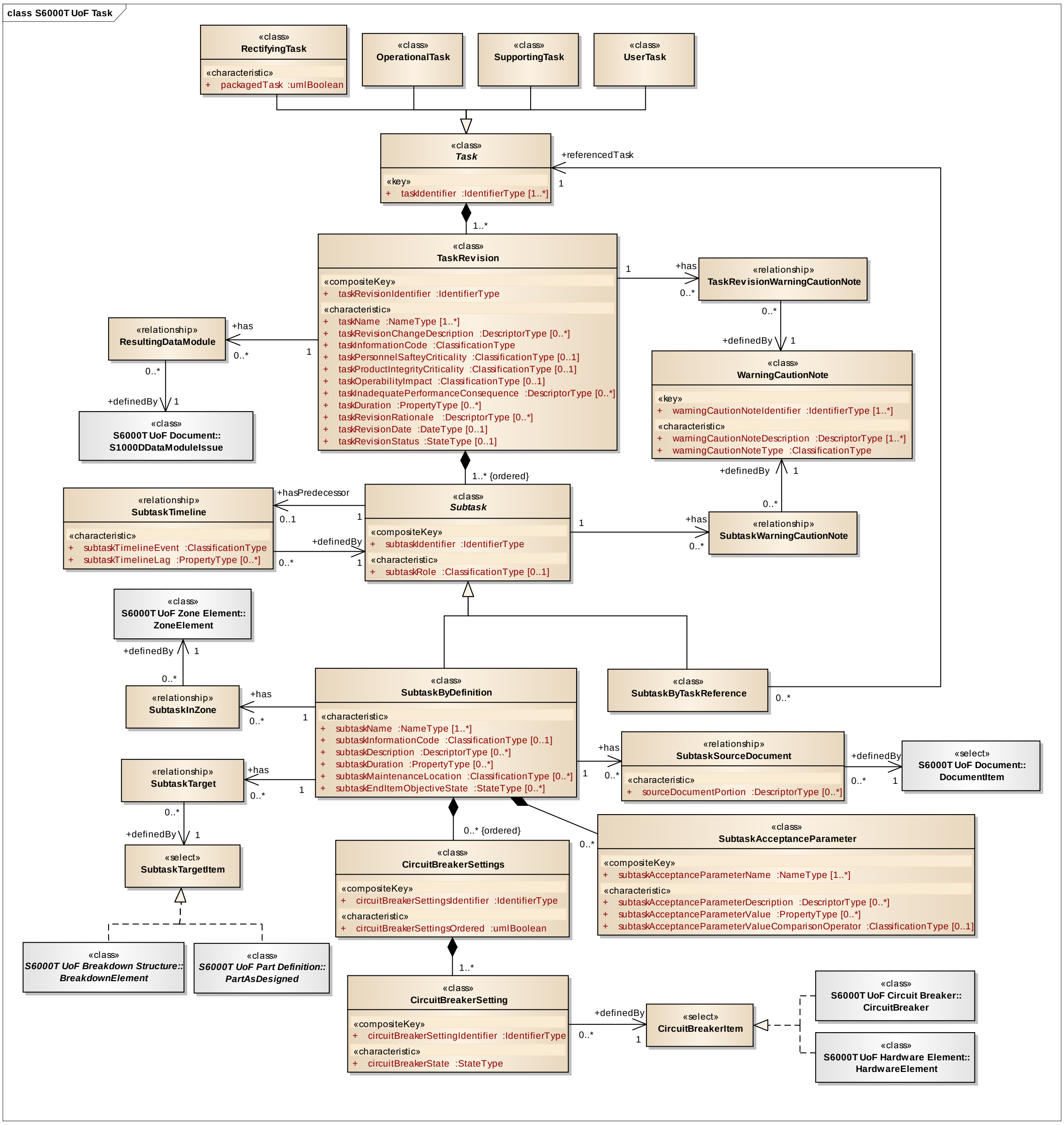
2 UoF Task

Key features of the UoF Task data model (refer to Fig 1) are:

- a task must be defined as either a rectifying task, operational task, supporting task or user task
- work steps within a task, are described using a set of subtasks
- a subtask can either be described in detail within the task under consideration (subtask by definition) or by a reference to another task (subtask by task reference)
- time dependencies between subtasks can be defined using subtask timeline

Note

A subtask, which is defined and described within a task, cannot be referenced by any other task.



ICN-S6000T-B6865-00035-002-01

Fig 1 UoF Task

2.1 CircuitBreakerItem

2.1.1 Interface definition

`CircuitBreakerItem` is a `<<select>>` interface that identifies items which can represent the referenced circuit breaker.

Source:

- S3000L

2.1.1.1 Class members

This <<select>> interface includes the following class members:

- CircuitBreaker. Refer to Chap 7.3.5.
- HardwareElement. Refer to Chap 7.3.13.

2.2 CircuitBreakerSetting

2.2.1 Class definition

CircuitBreakerSetting is a <<class>> that specifies an individual circuit breaker that must be in a specific state.

Source:

- SX001G

2.2.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of CircuitBreakerItem

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 circuitBreakerSettingIdentifier

circuitBreakerSettingIdentifier is an identifier that establishes a unique designator for a defined CircuitBreakerSetting, and to differentiate it from other instances of CircuitBreakerSetting.

Valid identifier class values:

- ID (SX001G:circuitBreakerSettingIdentifier)

Source:

- SX001G

2.2.2.2 circuitBreakerState

circuitBreakerState is a state that identifies the position that a given CircuitBreakerItem must be in after the accomplishment of a defined CircuitBreakerSetting.

Valid state values:

- C (SX001G:closeCircuitBreakerState)
- O (SX001G:openCircuitBreakerState)
- VC (SX001G:verifyCloseCircuitBreakerState)
- VO (SX001G:verifyOpenCircuitBreakerState)

Source:

– SX001G

2.3 CircuitBreakerSettings

2.3.1 Class definition

CircuitBreakerSettings is a <<class>> that identifies a set of circuit breakers that must be set in specific states.

Source:

– SX001G

2.3.1.1 Associations

This class has the following associations:

– An aggregate association with zero, one or many instances of CircuitBreakerSetting

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

– ApplicabilityStatementItem. Refer to Chap 7.3.2.
– DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
– ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
– RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 circuitBreakerSettingsIdentifier

circuitBreakerSettingsIdentifier is an identifier that establishes a unique designator for a defined set of circuit breaker settings, and to differentiate it from other instances of circuit breaker settings.

Valid identifier class values:

– ID (SX001G:circuitBreakerSettingsIdentifier)

Source:

– SX001G

2.3.2.2 circuitBreakerSettingsOrdered

circuitBreakerSettingsOrdered is a boolean that defines if the individual circuit breaker setting must be performed in the specified order.

Note

True specifies that the circuit breaker settings must be accomplished in the defined order. False specifies that the circuit breaker settings can be accomplished in any order.

Source:

– SX001G

2.4 OperationalTask

2.4.1 Class definition

OperationalTask is a Task that is required to support the use of a product.

Examples:

- fueling
- towing

Source:

- SX001G

2.4.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TaskRevision (inherited from Task)
- A directed has association with zero, one or many instances of TaskTarget (inherited from Task)

2.4.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem (inherited from Task). Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- EnvironmentDefinitionItem (inherited from Task). Refer to Chap 7.3.12.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from Task). Refer to Chap 7.3.28.

2.4.2 Attributes

2.4.2.1 taskIdentifier (one or many)

Inherited from Task.

2.5 RectifyingTask

2.5.1 Class definition

RectifyingTask is a Task that ensures or returns the function of the associated item.

Note

An event that can require a Task to be performed includes a failure, damage, a special event, and a time limit.

Examples:

- lubrication
- repair
- replace

Source:

- SX001G

2.5.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TaskRevision (inherited from Task)
- A directed has association with zero, one or many instances of TaskTarget (inherited from Task)

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem (inherited from Task). Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- EnvironmentDefinitionItem (inherited from Task). Refer to Chap 7.3.12.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from Task). Refer to Chap 7.3.28.

2.5.2 Attributes

2.5.2.1 taskIdentifier (one or many)

Inherited from Task.

2.5.2.2 packagedTask

packagedTask is a boolean that specifies if the Task is created to group a set of defined Tasks for a specific purpose.

Note

Grouping of Tasks that can be performed to support maintenance planning and scheduling activities.

Example:

- 1000 flight hours overhaul

Source:

- SX001G

2.6 ResultingDataModule

2.6.1 Class definition

ResultingDataModule is a <<relationship>> that identifies a data module issue in which a TaskRevision is further detailed.

Note

A data module can contain either a complete or partial description of a given task revision.

Source:

- SX001G

2.6.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of S1000DDataModuleIssue

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.7 Subtask

2.7.1 Class definition

Subtask is a <<class>> that represents the specification of a work step that is to be performed as part of a Task.

Source:

- SX001G

2.7.1.1 Associations

This class has the following associations:

- A directed hasPredecessor association with zero or one instance of SubtaskTimeline
- A directed has association with zero, one or many instances of SubtaskWarningCautionNote

2.7.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.7.2 Attributes

2.7.2.1 subtaskIdentifier

subtaskIdentifier is an identifier that establishes a unique designator for a Subtask and to differentiate it from other instances of Subtask.

Note

A Subtask is identified within the context of a specific Task.

Valid identifier class values:

- ID (SX001G:subtaskIdentifier)

Source:

- SX001G

2.7.2.2 subtaskRole (zero or one)

subtaskRole is a classification that identifies how the Subtask is related to the main function of the Task.

Note

SubtaskRole enables mapping between S3000L and the main portions of the S1000D procedure schema.

Valid Values:

- CL (SX001G:closeupSubtask)
- CON (SX001G:coreNoRequiredConditionsSubtask)
- COR (SX001G:coreSubtask)
- ST (SX001G:startupSubtask)

Source:

- SX001G

2.8 SubtaskAcceptanceParameter

2.8.1 Class definition

SubtaskAcceptanceParameter is a <<class>> that represents acceptance criteria which must be met before the Subtask is completed.

Source:

- S3000L

2.8.1.1 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.8.2 Attributes

2.8.2.1 subtaskAcceptanceParameterName (one or many)

subtaskAcceptanceParameterName is a name by which the SubtaskAcceptanceParameter is known and can be easily referenced.

Source:

- S3000L

2.8.2.2 subtaskAcceptanceParameterDescription (zero, one or many)

subtaskAcceptanceParameterDescription is a description that gives more information on the criteria that determines whether a Subtask is completed or not.

Source:

- S3000L

2.8.2.3 subtaskAcceptanceParameterValue (zero, one or many)

`subtaskAcceptanceParameterValue` is a property that specifies the value (criteria) that must be fulfilled before the `Subtask` can be ended.

Valid unit value:

– acceptanceParameterUnit

Source:

– S3000L

2.8.2.4 subtaskAcceptanceParameterValueComparisonOperator (zero or one)

`subtaskAcceptanceParameterValueComparisonOperator` is a classification that identifies the comparison operator against which a recorded value is to be evaluated against the `subtaskAcceptanceParameterValue`.

Valid values:

– `EQ` (SX001G:equalToComparisonOperator)
– `NE` (SX001G:notEqualToComparisonOperator)
– `LT` (SX001G:lessThanComparisonOperator)
– `LE` (SX001G:lessThanOrEqualToComparisonOperator)
– `GT` (SX001G:greaterThanComparisonOperator)
– `GE` (SX001G:greaterThanOrEqualToComparisonOperator)
– `IN` (SX001G:withinRangeComparisonOperator)
– `OUT` (SX001G:outsideRangeComparisonOperator)

Source:

– S3000L

2.9 SubtaskByDefinition

2.9.1 Class definition

`SubtaskByDefinition` is a `Subtask` that provides detailed information of the defined work step.

Source:

– SX001G

2.9.1.1 Associations

This class has the following associations:

– An aggregate association with zero, one or many instances of `CircuitBreakerSettings`
– An aggregate association with zero, one or many instances of `SubtaskAcceptanceParameter`
– A directed `hasPredecessor` association with zero or one instance of `SubtaskTimeline` (inherited from `Subtask`)
– A directed `has` association with zero, one or many instances of `SubtaskWarningCautionNote` (inherited from `Subtask`)
– A directed `has` association with zero, one or many instances of `SubtaskInZone`
– A directed `has` association with zero, one or many instances of `SubtaskSourceDocument`
– A directed `has` association with zero, one or many instances of `SubtaskTarget`

2.9.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from Subtask). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from Subtask). Refer to Chap 7.3.28.
- TaskResourceItem. Refer to Chap 7.3.35.

2.9.2 Attributes

2.9.2.1 subtaskIdentifier

Inherited from Subtask.

2.9.2.2 subtaskRole (zero or one)

Inherited from Subtask.

2.9.2.3 subtaskName (one or many)

subtaskName is a name by which the Subtask is known and can be easily referenced.

Source:

- SX001G

2.9.2.4 subtaskInformationCode (zero or one)

subtaskInformationCode is a classification that identifies the main purpose for the Subtask.

Note

An information code must be created in accordance with the rules defined in S1000D.

Valid values:

- Refer to the S6000T valid values XML schema for details

Source:

- SX001G

2.9.2.5 subtaskDescription (zero, one or many)

subtaskDescription is a description of the procedure performed during the Subtask.

Source:

- SX001G

2.9.2.6 subtaskDuration (zero, one or many)

subtaskDuration is a property that specifies the average time required for the performance of a Subtask, regardless of the number of personnel working simultaneously.

Note

subtaskDuration does not include time spent awaiting spares, support equipment, facilities, or personnel (logistics delay time).

Valid unit value:

- timeUnit

Source:

- SX001G

2.9.2.7 subtaskMaintenanceLocation (zero, one or many)

subtaskMaintenanceLocation is a classification that specifies where the Subtask will be performed in relation to the product.

Note

Proposed values equal the S1000D Item Location Codes.

Valid values:

- A (SX001G:maintenanceOnItemWhenInstalledOnProduct)
- B (SX001G:maintenanceOnItemOnMajorAssembly)
- C (SX001G:maintenanceOnBench)
- D (SX001G:maintenanceAnywhere)

Source:

- SX001G

2.9.2.8 subtaskEndItemObjectiveState (zero, one or many)

subtaskEndItemObjectiveState is a state that identifies the condition of the Product (refer to Chap 7.3.23) that will exist after the accomplishment of the Subtask.

Valid values:

- AS (SX001G:airSupplyEstablishedEndItemState)
- CS (SX001G:controlStatusEstablishedEndItemState)
- DF (SX001G:defueledEndItemState)
- EL (SX001G:electricalPowerEstablishedEndItemState)
- ELA (SX001G:electricalPowerFromAPUEstablishedEndItemState)
- ELE (SX001G:electricalPowerFromEngineEstablishedEndItemState)
- ELI (SX001G:internalElectricalPowerEstablishedEndItemState)
- ELX (SX001G:externalElectricalPowerEstablishedEndItemState)
- FU (SX001G:fueledEndItemState)
- HP (SX001G:hydraulicPowerEstablishedEndItemState)
- JC (SX001G:jackedEndItemState)
- SY (SX001G:safetyDeviceEstablishedEndItemState)
- TC (SX001G:taskCheckedEndItemState)
- UJ (SX001G:unjackedEndItemState)
- WS (SX001G:waterSupplyEstablishedEndItemState)

Source:

- SX001G

2.10 SubtaskByTaskReference

2.10.1 Class definition

SubtaskByTaskReference is a Subtask where the details of the subtask are defined as a separate Task.

Source:

- SX001G

2.10.1.1 Associations

This class has the following associations:

- A directed hasPredecessor association with zero or one instance of SubtaskTimeline (inherited from Subtask)
- A directed has association with zero, one or many instances of SubtaskWarningCautionNote (inherited from Subtask)
- A directed referencedTask association with one instance of Task

2.10.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from Subtask). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from Subtask). Refer to Chap 7.3.28.

2.10.2 Attributes

2.10.2.1 subtaskIdentifier

Inherited from Subtask.

2.10.2.2 subtaskRole (zero or one)

Inherited from Subtask.

2.11 SubtaskInZone

2.11.1 Class definition

SubtaskInZone is a <<relationship>> that identifies the zone where the Subtask is to be performed.

Source:

- SX001G

2.11.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of ZoneElement

2.11.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.12 SubtaskSourceDocument

2.12.1 Class definition

SubtaskSourceDocument is a <<relationship>> that identifies an external Document where the Subtask is originally defined, and more details are given.

Source:

- SX001G

2.12.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of DocumentItem

2.12.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.12.2 Attributes

2.12.2.1 sourceDocumentPortion (zero, one or many)

sourceDocumentPortion is a description that gives more information on relevant portions of the related Document. Refer to Chap 7.3.11.

Source:

- SX001G

2.13 SubtaskTarget

2.13.1 Class definition

SubtaskTarget is a <<relationship>> that identifies the item on which the Subtask is to be performed.

Source:

- SX001G

2.13.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of SubtaskTargetItem

2.13.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.14 SubtaskTargetItem

2.14.1 Interface definition

SubtaskTargetItem is a <<select>> interface that identifies items on which a Subtask can be performed.

Source:

- SX001G

2.14.1.1 Class members

This <<select>> interface includes the following class members:

- BreakdownElement. Refer to Chap 7.3.3.
- PartAsDesigned. Refer to Chap 7.3.21.

2.15 SubtaskTimeline

2.15.1 Class definition

SubtaskTimeline is a <<relationship>> that identifies that there is a time dependency between two Subtasks within the same Task.

Source:

- SX001G

2.15.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of Subtask

2.15.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.15.2 Attributes

2.15.2.1 subtaskTimelineEvent

subtaskTimelineEvent is a classification that identifies how the starting point for a Subtask depends upon its preceding Subtask.

Valid values:

- END (SX001G:subtaskEndEvent)
- START (SX001G:subtaskStartEvent)

Source:

- SX001G

2.15.2.2 subtaskTimelineLag (zero, one or many)

subtaskTimelineLag is a property that specifies the time that must elapse before the Subtask under consideration can start, in relation to its associated timeline event.

Valid unit value:

- timeUnit

Source:

- SX001G

2.16 SubtaskWarningCautionNote

2.16.1 Class definition

SubtaskWarningCautionNote is a <<relationship>> that identifies a WarningCautionNote that is associated with a given Subtask.

Source:

- SX001G

2.16.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of WarningCautionNote

2.16.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.17 SupportingTask

2.17.1 Class definition

SupportingTask is a Task that does not meet a TaskRequirement, but identifies a set of work steps which will be carried out as part of multiple Tasks.

Note

A SupportingTask will only be used in the context of SubtaskByTaskReference.

Note

The objective for a SupportingTask is to enable reuse of a sequence of work steps, needed by a set of Tasks.

Examples:

- jack vehicle
- open hatch

Source:

- SX001G

2.17.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TaskRevision (inherited from Task)
- A directed has association with zero, one or many instances of TaskTarget (inherited from Task)

2.17.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem (inherited from Task). Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- EnvironmentDefinitionItem (inherited from Task). Refer to Chap 7.3.12.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from Task). Refer to Chap 7.3.28.

2.17.2 Attributes

2.17.2.1 taskIdentifier (one or many)

Inherited from Task.

2.18 Task

2.18.1 Class definition

Task is a <<class>> that represents the specification of work to be done or undertaken.

Source:

- SX001G

2.18.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TaskRevision
- A directed has association with zero, one or many instances of TaskTarget

2.18.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem. Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- EnvironmentDefinitionItem. Refer to Chap 7.3.12.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem. Refer to Chap 7.3.28.

2.18.2 Attributes

2.18.2.1 taskIdentifier (one or many)

taskIdentifier is an identifier that establishes a unique designator for a Task and to differentiate it from other instances of Task.

Valid identifier class values:

– ID (SX001G:taskIdentifier)

Source:

– SX001G

2.19 TaskRevision

2.19.1 Class definition

TaskRevision is a <<class>> representing an iteration applied to a Task.

Source:

– SX001G

2.19.1.1 Associations

This class has the following associations:

- An ordered aggregate association with one or many instances of Subtask
- An aggregate association with zero, one or many instances of TaskPerformanceObjective
- An aggregate association with zero, one or many instances of TaskTrainDecision
- A directed has association with zero, one or many instances of ResultingDataModule
- A directed has association with zero, one or many instances of TaskRevisionWarningCautionNote

2.19.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ChangeControlledItem. Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- EnvironmentDefinitionItem. Refer to Chap 7.3.12.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- TaskResourceItem. Refer to Chap 7.3.35.

2.19.2 Attributes

2.19.2.1 taskRevisionIdentifier

taskRevisonIdentifier is an identifier that establishes a unique designator for a TaskRevision and to differentiate it from other instances of TaskRevision.

Valid identifier class values:

– ID (SX001G:taskRevisonIdentifier)

Source:

– SX001G

2.19.2.2 taskName (one or many)
taskName is a name by which the Task is known and can be easily referenced.

Source:

– SX001G

2.19.2.3 taskRevisionChangeDescription (zero, one or many)
taskRevisionChangeDescription is a description that gives more information on content that has been altered between two revisions of a Task.

Source:

– SX001G

2.19.2.4 taskInformationCode
taskInformationCode is a classification that identifies the main purpose for the Task.

Note
An information code must be created in accordance with the rules defined in S1000D.

Valid values:

– Refer to the S6000T valid values XML schema for details

Source:

– SX001G

2.19.2.5 taskPersonnelSafteyCriticality (zero or one)
taskPersonnelSafteyCriticality is a classification that identifies the most serious health aspects that the performance of the Task can pose on personnel performing the Task.

Valid values:

– 1 (SX001G:catastrophicPersonnelSafetyCriticality)
– 2 (SX001G:criticalPersonnelSafetyCriticality)
– 3 (SX001G:marginalPersonnelSafetyCriticality)
– 4 (SX001G:negligiblePersonnelSafetyCriticality)

Source:

– S3000L

2.19.2.6 taskProductIntegrityCriticality (zero or one)
taskProductIntegrityCriticality is a classification that identifies whether the Task is critical (ie, if failure to accomplish it would result in adverse effects on product or system reliability, efficiency, effectiveness, safety, or cost).

Note
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 values:

- N (SX001G:nonCriticalProductIntegrityTask)
- Y (SX001G:criticalProductIntegrityTask)

Source:

- S3000L

2.19.2.7 taskOperabilityImpact (zero or one)
taskOperabilityImpact is a classification that indicates the operational status and mission readiness of the end item during the Task.

Valid values:

- A (SX001G:systemInoperableDuringTaskExecution)
- B (SX001G:systemOperableDuringTaskExecution)
- C (SX001G:systemFullMissionCapableDuringTaskExecution)
- D (SX001G:systemPartialMissionCapableDuringTaskExecution)
- E (SX001G:productNotMissionCapableDuringTaskExecution)
- G (SX001G:turnaroundTask)

Source:

- S3000L

2.19.2.8 taskInadequatePerformanceConsequence (zero, one or many)
taskInadequatePerformanceConsequence is a description that provides further information about undesired adverse reliability, efficiency, effectiveness, safety, or cost effects related to the Product (refer to Chap 7.3.23) or system if the accomplishment of the Task fails.

2.19.2.9 taskDuration (zero, one or many)
taskDuration is a property that specifies the average time required for the performance of a Task, regardless of the number of personnel working simultaneously.

Note
taskDuration does not include time spent awaiting spares, support equipment, facilities, or personnel (logistics delay time).

Note
taskDuration could be calculated from the subtask durations.

Valid unit value:

- timeUnit

Source:

- SX001G

2.19.2.10 taskRevisionRationale (zero, one or many)
taskRevisionRationale is a description that gives more information on the justification for revising the Task.

Source:

- SX001G

2.19.2.11 taskRevisionDate (zero or one)

taskRevisionDate is a date that specifies when the Task was revised.

Source:

- SX001G

2.19.2.12 taskRevisionStatus (zero or one)

taskRevisionStatus is a state that identifies the progress on the development of a TaskRevision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

Source:

- SX001G

2.20 TaskRevisionWarningCautionNote

2.20.1 Class definition

TaskRevisionWarningCautionNote is a <<relationship>> that identifies a WarningCautionNote that is associated with a given Task.

Source:

- SX001G

2.20.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of WarningCautionNote

2.20.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.21 UserTask

2.21.1 Class definition

UserTask is a Task that defines a required interaction between the user of a Product (refer to Chap 7.3.23) and the Product itself.

Examples:

- land an aircraft
- park a car

2.21.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TaskRevision (inherited from Task)
- A directed has association with zero, one or many instances of TaskTarget (inherited from Task)

2.21.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DigitalFileReferencingItem (inherited from Task). Refer to Chap 7.3.10.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- EnvironmentDefinitionItem (inherited from Task). Refer to Chap 7.3.12.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from Task). Refer to Chap 7.3.28.

2.21.2 Attributes

2.21.2.1 taskIdentifier (one or many)

Inherited from Task.

2.22 WarningCautionNote

2.22.1 Class definition

WarningCautionNote is a <<class>> that defines advice concerning safety, legal and health aspects.

Source:

- SX001G

2.22.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of WarningCautionTrainDecision (refer to Chap 7.3.45)

- Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.22.2 Attributes

2.22.2.1 warningCautionNoteIdentifier (one or many)

warningCautionNoteIdentifier is an identifier that establishes a unique designator for a WarningCautionNote, and to differentiate it from other instances of WarningCautionNote.

Valid identifier class values:

- ID (SX001G:warningCautionNoteIdentifier)

Source:

- SX001G

2.22.2.2 warningCautionNoteDescription (one or many)
`warningCautionNoteDescription` is a description that gives more information on safety, legal and health considerations.

Source:

- SX001G

2.22.2.3 warningCautionNoteType
`warningCautionNoteType` is a classification that identifies severity and scope for the safety, legal and health considerations.

Valid values:

- C (SX001G:cautionAdvise)
- N (SX001G:noteAdvise)
- W (SX001G:warningAdvise)

Source:

- SX001G

Chapter 7.3.33

Units of functionality - UoF Task Knowledge Skill and Attitude

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark

Chap No./Document No.	Title
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization
Chap 7.3.45	Units of functionality - UoF Warning Caution Train Prioritization
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

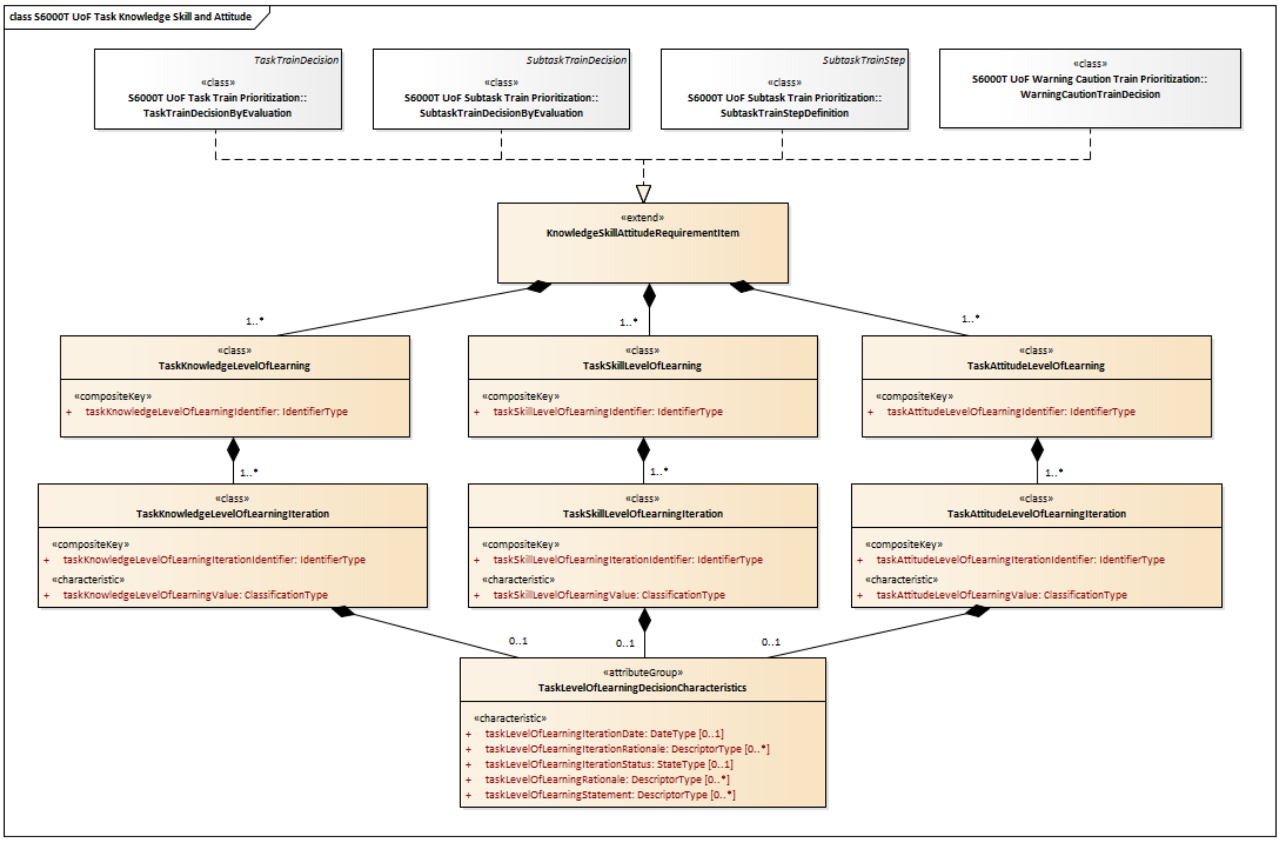
1 General

The Task Knowledge Skill and Attitude UoF supports the identification of knowledge, skill and attitude required to perform a task, subtask, or subtask step.

2 UoF Task Knowledge Skill and Attitude

Key features of the UoF Task Knowledge Skill and Attitude data model, (refer to Fig 1), are:

- Level of learning must be defined for both the knowledge (cognitive), skill (psychomotor) and attitude (affective) domain
- Decision on required level of learning can be iterated per domain. For example, feedback on actual performance in the field can result in iterating the decision on required level of learning.



ICN-S6000T-B6865-00046-003-01

Fig 1 UoF Task Knowledge Skill and Attitude

2.1 KnowledgeSkillAttitudeRequirementItem

2.1.1 Interface definition

KnowledgeSkillAttitudeRequirementItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

2.1.1.1 Associations

The KnowledgeSkillAttitudeRequirementItem <<extend>> interface has the following associations:

- An aggregate association with one or many instances of TaskAttitudeLevelOfLearning
- An aggregate association with one or many instances of TaskKnowledgeLevelOfLearning
- An aggregate association with one or many instances of TaskSkillLevelOfLearning

2.1.1.2 Class members

Classes that implement the KnowledgeSkillAttitudeRequirementItem <<extend>> interface are:

- SubtaskTrainDecisionByEvaluation. Refer to Chap 7.3.30.
- SubtaskTrainStepDefinition. Refer to Chap 7.3.30.
- TaskTrainDecisionByEvaluation. Refer to Chap 7.3.37.
- WarningCautionTrainDecision. Refer to Chap 7.3.45.

2.2 TaskAttitudeLevelOfLearning

2.2.1 Class definition

TaskAttitudeLevelOfLearning is a <<class>> that defines the task's level of sophistication with respect to emotional, motivational and/or social aspects.

Examples:

- Appreciation
- Enthusiasm
- Feelings
- Motivation
- Values

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of TaskAttitudeLevelOfLearningIteration

2.2.1.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 taskAttitudeLevelOfLearningIterationIdentifier

taskAttitudeLevelOfLearningIterationIdentifier is an identifier that establishes a unique designator for a TaskAttitudeLevelOfLearning iteration and to differentiate it from other instances of TaskAttitudeLevelOfLearning iteration.

Valid identifier class values:

- ID (SX001G:taskAttitudeLevelOfLearningIterationIdentifier)

2.2.2.2 taskAttitudeLevelOfLearningValue

taskAttitudeLevelOfLearningValue is a classification that defines the level of emotional, motivational, and social sophistication that a person must accomplish.

Valid values:

- Refer to the S6000T valid values XML schema for details

2.3 TaskKnowledgeLevelOfLearning

2.3.1 Class definition

TaskKnowledgeLevelOfLearning is a <<class>> that defines the task's level of sophistication with respect to mental, cognitive, and/or logical aspects.

Examples:

- Create knowledge
- Process information
- Thinking

2.3.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of TaskKnowledgeLevelOfLearningIteraction

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 taskKnowledgeLevelOfLearningIterationIdentifier

taskKnowledgeLevelOfLearningIterationIdentifier is an identifier that establishes a unique designator for a TaskKnowledgeLevelOfLearning iteration and to differentiate it from other instances of TaskKnowledgeLevelOfLearning iteration.

Valid identifier class values:

- ID (SX001G:taskKnowledgeLevelOfLearningIterationIdentifier)

2.3.2.2 taskKnowledgeLevelOfLearningValue

`taskKnowledgeLevelOfLearningValue` is a classification that defines the mental and logical sophistication that a person must be able to accomplish.

Valid values:

- Refer to the S6000T valid values XML schema for details

2.4 TaskLevelOfLearningDecisionCharacteristics

2.4.1 Attribute group definition

`TaskLevelOfLearningDecisionCharacteristics` is an `<<attributeGroup>>` that provides the capability to record additional information for the required task level of learning decision and iteration.

2.4.2 Attributes

2.4.2.1 taskLevelOfLearningStatement (zero, one or many)

`taskLevelOfLearningDescription` is a description that gives more information on the required level of learning.

2.4.2.2 taskLevelOfLearningRationale (zero, one or many)

`taskLevelOfLearningRationale` is a description that gives more information on the justification for the chosen level of learning.

2.4.2.3 taskLevelOfLearningIterationRationale (zero, one or many)

`taskLevelOfLearningIterationRationale` is a description that gives more information on the justification to revise the defined level of learning for the associated task, subtask or subtask step.

2.4.2.4 taskLevelOfLearningIterationDate (zero or one)

`taskLevelOfLearningIterationDate` is a date that specifies when the defined level of learning for the associated task, subtask or subtask step was revised.

2.4.2.5 taskLevelOfLearningIterationStatus (zero or one)

`taskLevelOfLearningIterationStatus` is a state that identifies the maturity of the task level of learning iteration.

Valid state values:

- `A` (SX001G:approvedStatus)
- `D` (SX001G:draftStatus)
- `IW` (SX001G:inWorkStatus)
- `C` (SX001G:cancelledStatus)
- `R` (SX001G:reviewedStatus)

2.5 TaskSkillLevelOfLearning

2.5.1 Class definition

`TaskSkillLevelOfLearning` is a `<<class>>` that defines the task's level of sophistication with respect to physical, mechanical, and/or movement-based aspects.

Examples:

- Coordination
- Motor skills
- Physical movement

2.5.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of TaskSkillLevelOfLearningIteraction

2.5.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 taskSkillLevelOfLearningIterationIdentifier

taskSkillLevelOfLearningIterationIdentifier is an identifier that establishes a unique designator for a TaskSkillLevelOfLearning iteration and to differentiate it from other instances of TaskSkillLevelOfLearning iteration.

Valid identifier class values:

- ID (SX001G:taskSkillLevelOfLearningIterationIdentifier)

2.5.2.2 taskSkillLevelOfLearningValue

taskSkillLevelOfLearningValue is a classification that defines the physical, mechanical, and movement-based sophistication that a person must accomplish.

Valid values:

- Refer to the S6000T valid values XML schema for details

2.6 TaskKnowledgeLevelOfLearningIteration

2.6.1 Class definition

TaskKnowledgeLevelOfLearningIteration is a <<class>> representing an iteration applied to a TaskKnowledgeLevelOfLearning.

2.6.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TaskKnowledgeLevelOfLearning

2.6.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 taskKnowledgeLevelOfLearningIterationIdentifier (one or many)

taskKnowledgeLevelOfLearningIterationIdentifier is an identifier that establishes a unique designator for a TaskKnowledgeLevelOfLearning iteration and to differentiate it from other instances of TaskKnowledgeLevelOfLearning iteration.

2.6.2.2 taskKnowledgeLevelOfLearningValue

taskKnowledgeLevelOfLearningValue is a classification that defines the mental and logical sophistication that a person must be able to accomplish.

2.7 TaskSkillLevelOfLearningIteration

2.7.1 Class definition

TaskSkillLevelOfLearningIteration is a <<class>> representing an iteration applied to a TaskSkillLevelOfLearning.

2.7.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TaskSkillLevelOfLearning

2.7.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.7.2 Attributes

2.7.2.1 taskSkillLevelOfLearningIterationIdentifier (one or many)

taskSkillLevelOfLearningIterationIdentifier is an identifier that establishes a unique designator for a TaskSkillLevelOfLearning iteration and to differentiate it from other instances of TaskSkillLevelOfLearning iteration.

2.7.2.2 taskSkillLevelOfLearningValue

taskSkillLevelOfLearningValue is a classification that defines the physical, mechanical, and movement-based sophistication that a person must accomplish.

2.8 TaskAttitudeLevelOfLearningIteration

2.8.1 Class definition

TaskAttitudeLevelOfLearningIteration is a <<class>> representing an iteration applied to a TaskAttitudeLevelOfLearning.

2.8.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TaskAttitudeLevelOfLearning

2.8.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.8.2 Attributes

2.8.2.1 taskAttitudeLevelOfLearningIterationIdentifier (one or many)

taskAttitudeLevelOfLearningIterationIdentifier is an identifier that establishes a unique designator for a TaskAttitudeLevelOfLearning iteration and to differentiate it from other instances of TaskAttitudeLevelOfLearning iteration.

2.8.2.2 taskAttitudeLevelOfLearningValue

taskAttitudeLevelOfLearningValue is a classification that defines the level of emotional, motivational, and social sophistication that a person must accomplish.

Chapter 7.3.34

Units of functionality - UoF Task Performance Objective

Table of contents

Page

List of tables

List of figures

References

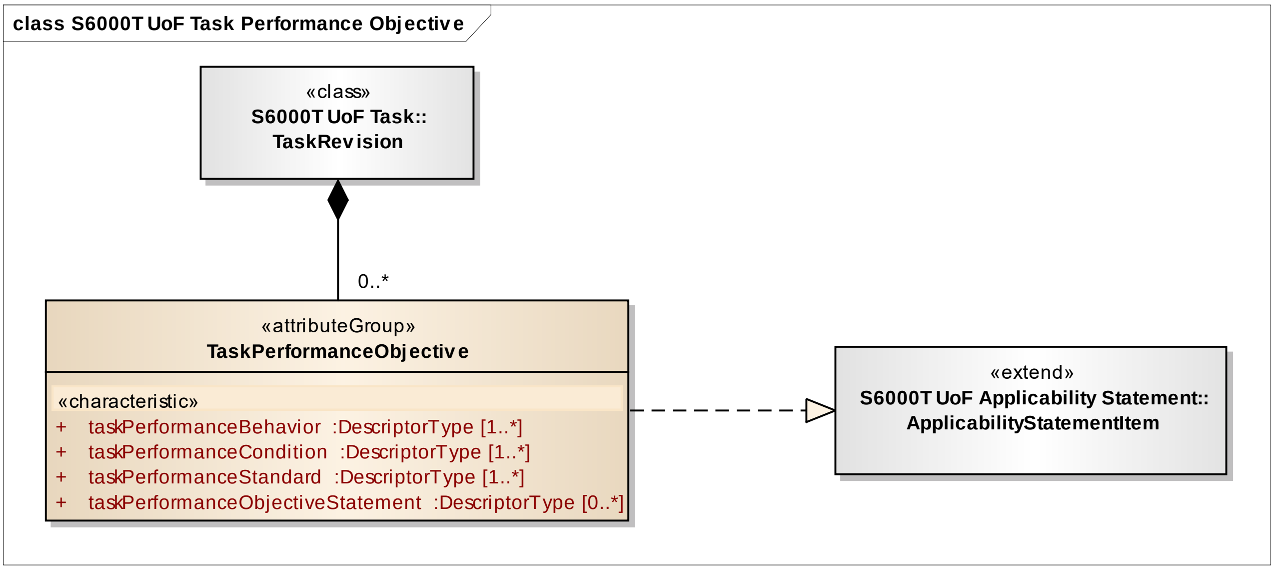
Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.32	Units of functionality - UoF Task
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Task Performance Objective UoF supports the definition of required behavior and performance standards that the person executing a specific task must meet, given the conditions under which the task is to be performed.

2 UoF Task Performance Objective



ICN-S6000T-B6865-00042-002-01

Fig 1 UoF Task Performance Objective

2.1 TaskPerformanceObjective

2.1.1 Attribute group definition

TaskPerformanceObjective is an <<attributeGroup>> that defines the precise statement of the performance expected in the execution of a task expressed in terms of the task performance conditions, behavior and standard.

2.1.1.1 Implementations

This attribute group implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.

2.1.2 Attributes

2.1.2.1 taskPerformanceBehavior (one or many)

taskPerformanceBehavior is a description that expresses the main purpose for the Task. Refer to Chap 7.3.32.

2.1.2.2 taskPerformanceCondition (one or many)

taskPerformanceCondition is a description that expresses the environmental, physical, and psychological circumstances under which the Task (refer to Chap 7.3.32) will be performed.

2.1.2.3 taskPerformanceStandard (one or many)

taskPerformanceStandard is a description that expresses the expected quality and/or time constraints that must be met in the performance of the Task (refer to Chap 7.3.32).

2.1.2.4 taskPerformanceObjectiveStatement (zero, one or many)

taskPerformanceObjectiveStatement is a description that expresses the taskPerformanceBehavior, taskPerformanceCondition and the taskPerformanceStandard in one sentence.

Chapter 7.3.35

Units of functionality - UoF Task Resource

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.11	Units of functionality - UoF Document

Chap No./Document No.	Title
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.32	Units of functionality - UoF Task
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

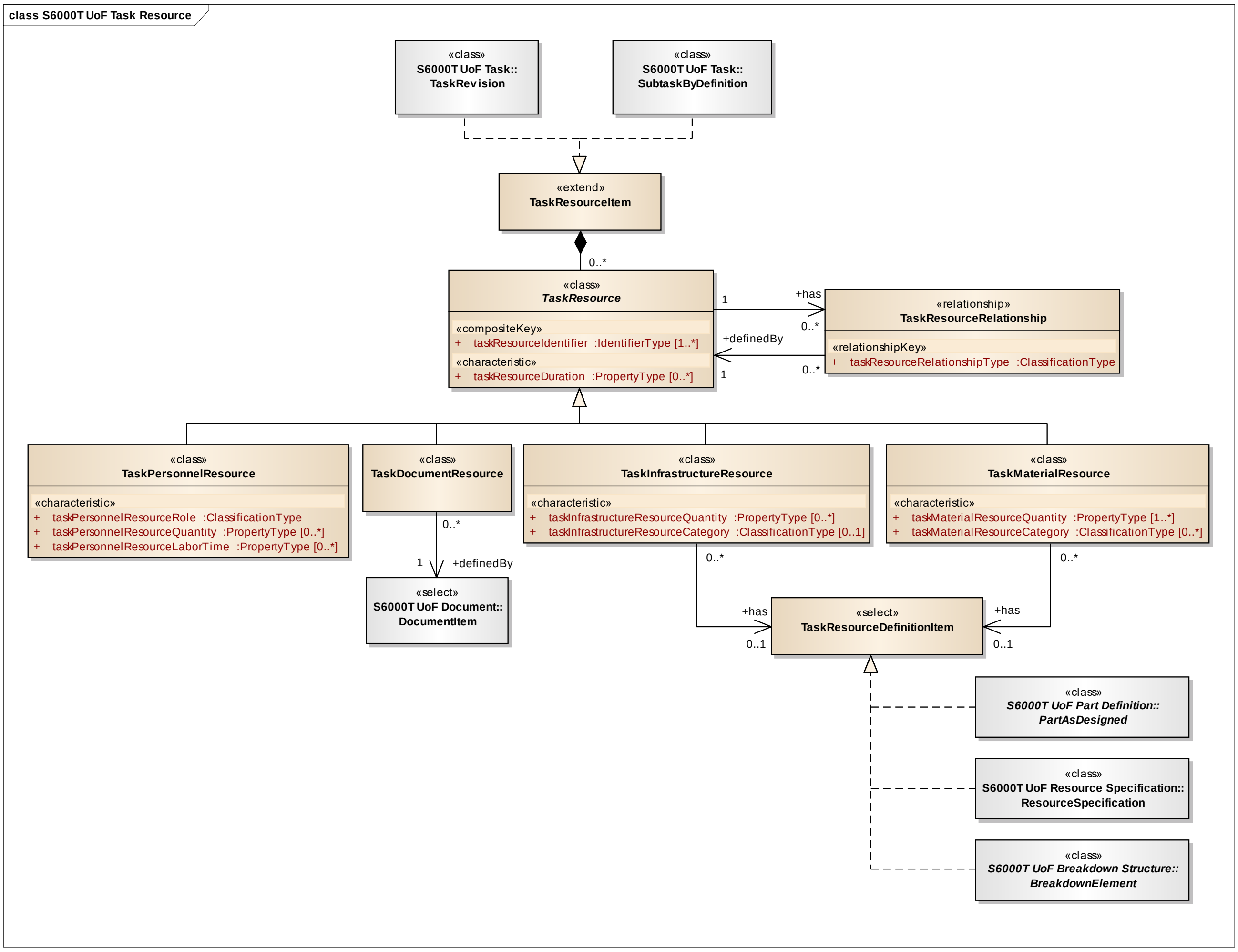
1 General

The Task Resource UoF supports a detailed specification of resources needed to perform a specified amount of work.

2 UoF Task Resource

Key features of the UoF Task Resource data model, (refer to Fig 1), are:

- a task resource can either be specified for the overall task or per subtask
- a task resource must be defined as either:
  - a material resource (eg, spare part, tool, consumable, etc)
  - an infrastructure resource (eg, hangar, power, etc)
  - a personnel resource (eg, skill, trade, etc)
  - a document resource (eg, forms for recording test results, etc)



ICN-S6000T-B6865-00036-002-01

Fig 1 UoF Task Resource

2.1 TaskDocumentResource

2.1.1 Class definition

`TaskDocumentResource` is a `TaskResource` that identifies a `Document` (refer to Chap 7.3.11) used as a resource.

Example:

– a form that must be filled out before, during or after the specified amount of work is carried out.

Source:

– SX001G

2.1.1.1 Associations

This class has the following associations:

– A directed `has` association with zero, one or many instances of `TaskResourceRelationship` (inherited from `TaskResource`)
– A directed `definedBy` association with one instance of `DocumentItem`

2.1.1.2 Implementations
This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from TaskResource). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 taskResourceIdentifier (one or many)
Inherited from TaskResource.

2.1.2.2 taskResourceDuration (zero, one or many)
Inherited from TaskResource.

2.2 TaskInfrastructureResource

2.2.1 Class definition

TaskInfrastructureResource is a TaskResource that defines foundational systems and services required as a resource.

Source:

- SX001G

2.2.1.1 Associations
This class has the following associations:

- A directed has association with zero, one or many instances of TaskResourceRelationship (inherited from TaskResource)
- A directed has association with zero or one instance of TaskResourceDefinitionItem

2.2.1.2 Implementations
This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from TaskResource). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 taskResourceIdentifier (one or many)
Inherited from TaskResource.

2.2.2.2 taskResourceDuration (zero, one or many)
Inherited from TaskResource.

2.2.2.3 taskInfrastructureResourceQuantity (zero, one or many)
taskInfrastructureResourceQuantity is a property that specifies the number of the TaskInfrastructureResource.

Valid unit value:

– quantityUnit

Source:

– SX001G

2.2.2.4 taskInfrastructureResourceCategory (zero or one)
taskInfrastructureResourceCategory is a classification that identifies further specialization for a TaskInfrastructureResource.

Valid values:

– CNWK (SX001G:communicationNetworkInfrastructureResource)
– COMP (SX001G:computerInfrastructureResource)
– DOCK (SX001G:dockInfrastructureResource)
– DRYD (SX001G:dryDockInfrastructureResource)
– GAR (SX001G:garageInfrastructureResource)
– HNG (SX001G:hangarInfrastructureResource)
– POW (SX001G:powerInfrastructureResource)
– TNWK (SX001G:transportNetworkInfrastructureResource)

Source:

– SX001G

2.3 TaskMaterialResource

2.3.1 Class definition

TaskMaterialResource is a TaskResource that identifies parts which are required as a resource.

Source:

– SX001G

2.3.1.1 Associations
This class has the following associations:

– A directed has association with zero, one or many instances of TaskResourceRelationship (inherited from TaskResource)
– A directed has association with zero or one instance of TaskResourceDefinitionItem

2.3.1.2 Implementations
This class implements the following <<extend>> interfaces:

– ApplicabilityStatementItem (inherited from TaskResource). Refer to Chap 7.3.2.
– DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.

- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 taskResourceIdentifier (one or many)

Inherited from TaskResource.

2.3.2.2 taskResourceDuration (zero, one or many)

Inherited from TaskResource.

2.3.2.3 taskMaterialResourceQuantity (one or many)

taskMaterialResourceQuantity is a property that specifies the number of a TaskMaterialResource.

Valid unit value:

- quantityUnit

Source:

- SX001G

2.3.2.4 taskMaterialResourceCategory (zero, one or many)

taskMaterialResourceCategory is a classification that defines the role of the TaskMaterialResource in the context of the specified Task (refer to Chap 3.7.32).

Valid values:

- C (SX001G:consumablePart)
- E (SX001G:expendableSparePart)
- EP (SX001G:expendablePersonalProtectionPart)
- HT (SX001G:standardHandTool)
- IT (SX001G:informationTechnologyPart)
- PA (SX001G:packagingPart)
- PE (SX001G:personalProtectionPart)
- PR (SX001G:productProtectionPart)
- R (SX001G:repairableSparePart)
- M (SX001G:rawMaterial)
- S (SX001G:sparePart)
- SE (SX001G:supportEquipment)
- SSE (SX001G:safetyRelatedSupportEquipment)

Source:

- SX001G

2.4 TaskPersonnelResource

2.4.1 Class definition

TaskPersonnelResource is a TaskResource that specifies the manpower required as a resource.

Source:

- SX001G

2.4.1.1 Associations

This class has the following associations:

- A directed `has` association with zero, one or many instances of `TaskResourceRelationship` (inherited from `TaskResource`)

2.4.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `ApplicabilityStatementItem` (inherited from `TaskResource`). Refer to Chap 7.3.2.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 taskResourceIdentifier (one or many)

Inherited from `TaskResource`.

2.4.2.2 taskResourceDuration (zero, one or many)

Inherited from `TaskResource`.

2.4.2.3 taskPersonnelResourceRole

`taskPersonnelResourceRole` is a classification that defines the purpose of the required `TaskPersonnelResource`.

Note

Traditional use of eg, Man A and Man B to identify personnel resources can qualify as `taskResourceIdentifiers` in the S-Series IPS specifications.

Examples:

- performer
- quality assurance
- supervisor

Valid values:

- `A` (SX001G:assistingTaskPersonnelResource)
- `P` (SX001G:performerTaskPersonnelResource)
- `Q` (SX001G:qualityAssuranceTaskPersonnelResource)
- `S` (SX001G:supervisorTaskPersonnelResource)

Source:

- SX001G

2.4.2.4 taskPersonnelResourceQuantity (zero, one or many)

`taskPersonnelResourceQuantity` is a property that specifies the number of the required `TaskPersonnelResource`

Note

Quantities can be given as single values but also as ranges or as text, (eg, "AsRequired").

Valid unit value:

- countUnit

Source:

- SX001G

2.4.2.5 taskPersonnelResourceLaborTime (zero, one or many)
`taskPersonnelResourceLaborTime` is a property that specifies the time expended by the required `TaskPersonnelResource`.

Note
Labor time can be given as single values but also as ranges or as text, (eg, "AsRequired").

Valid unit value:

- timeConsumptionUnit

Source:

- SX001G

2.5 TaskResource

2.5.1 Class definition

`TaskResource` is a `<<class>>` that identifies means that must be available to perform a specified amount of work.

Source:

- SX001G

2.5.1.1 Associations
This class has the following associations:

- A directed `has` association with zero, one or many instances of `TaskResourceRelationship`

2.5.1.2 Implementations
This class implements the following `<<extend>>` interfaces:

- `ApplicabilityStatementItem`. Refer to Chap 7.3.2.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.5.2 Attribute

2.5.2.1 taskResourceIdentifier (one or many)
`taskResourceIdentifier` is an identifier that establishes a unique designator for a `TaskResource` and to differentiate it from other instances of `TaskResource`.

Valid identifier class values:

- `ID` (SX001G:taskResourceIdentifier)

Source:

- SX001G

2.5.2.2 taskResourceDuration (zero, one or many)
taskResourceDuration is a property that specifies the average time that a TaskResource is needed to perform a specified amount of work.

Valid unit value:

- timeUnit

Source:

- SX001G

2.6 TaskResourceDefinitionItem

2.6.1 Interface definition

TaskResourceDefinitionItem is a <<select>> interface that identifies which items can be used as either infrastructure or material resources.

Source:

- SX001G

2.6.1.1 Class members
This <<select>> interface includes the following class members:

- BreakdownElement. Refer to Chap 7.3.3.
- PartAsDesigned. Refer to Chap 7.3.21.
- ResourceSpecification. Refer to Chap 7.3.27.

2.7 TaskResourceItem

2.7.1 Interface definition

TaskResourceItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.7.1.1 Associations
The TaskResourceItem <<extend>> interface has the following associations:

- An aggregate association with zero, one or many instances of TaskResource

2.7.1.2 Class members
Classes that implement the TaskResourceItem <<extend>> interface are:

- SubtaskByDefinition. Refer to Chap 7.3.32.
- TaskRevision. Refer to Chap 7.3.32.

2.8 TaskResourceRelationship

2.8.1 Class definition

TaskResourceRelationship is a <<relationship>> where one TaskResource relates to another TaskResource.

Note

Instances of task resource that are related to each other must be defined as resources for the same instance of task or subtask.

Example:

- person A "uses" tool B

Source:

- SX001G

2.8.1.1 Associations

This class has the following associations:

- A directed `definedBy` association with one instance of `TaskResource`

2.8.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

- `ApplicabilityStatementItem`. Refer to Chap 7.3.2.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.8.2 Attributes

2.8.2.1 taskResourceRelationshipType

`taskResourceRelationshipType` is a classification that identifies the meaning of the established relationship.

Valid values:

- `AND` (SX001G:mutualInclusionTaskResourceRelationship)
- `COOP` (SX001G:cooperationTaskResourceRelationship)
- `SEQ` (SX001G:sequenceTaskResourceRelationship)
- `USE` (SX001G:usesTaskResourceRelationship)
- `XOR` (SX001G:mutualExclusionTaskResourceRelationship)

Source:

- SX001G

Chapter 7.3.36

Units of functionality - UoF Task Target

Table of contents

Page

List of tables

List of figures

References

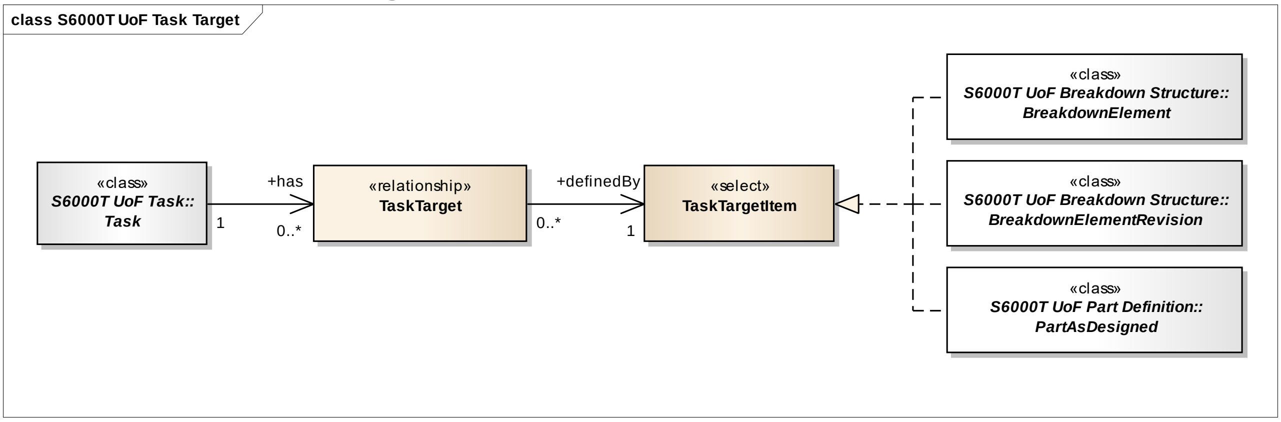
Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.26	Units of functionality - UoF Remark
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Task Target UoF provides the capability to identify items on which a task will be performed.

2 UoF Task Target



ICN-S6000T-B6865-00097-001-01

Fig 1 UoF Task Target

2.1 TaskTarget

2.1.1 Class definition

TaskTarget is a <<relationship>> that identifies the TaskTargetItem on which the Task is to be performed.

2.1.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of TaskTargetItem

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2 TaskTargetItem

2.2.1 Interface definition

TaskTargetItem is a <<select>> interface that identifies items on which a Task can be performed.

2.2.1.1 Class members

This <<select>> interface includes the following class members:

- BreakdownElement. Refer to Chap 7.3.3.
- BreakdownElementRevision. Refer to Chap 7.3.3.
- PartAsDesigned. Refer to Chap 7.3.21.

Chapter 7.3.37

Units of functionality - UoF Task Train Prioritization

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

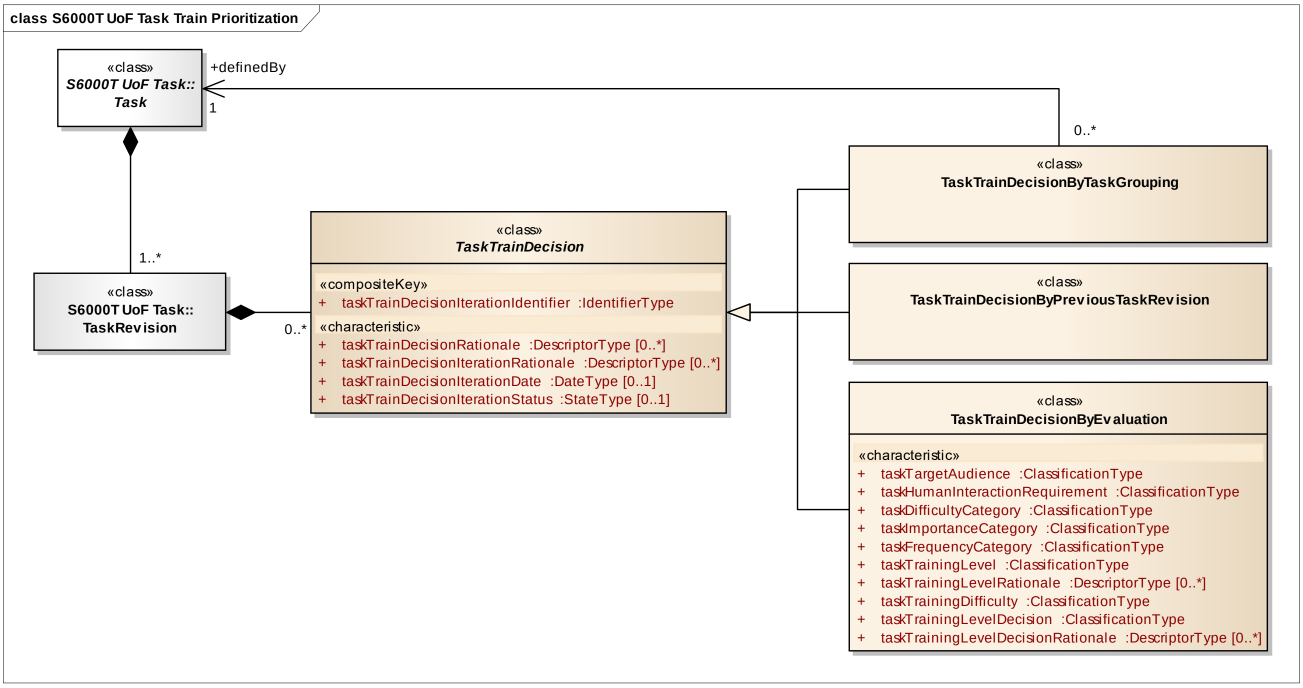
1 General

The Task Train Prioritization UoF provides the capability to define training prioritization for a task.

2 UoF Task Train Prioritization

Key features of the UoF Task Train Prioritization data model, (refer to Fig 1), are:

- task train prioritization for a task and its revision can be grouped by allowing tasks to refer to other tasks for its train prioritization. For example, a task for remove left front tire on a car can refer to the task for remove right front tire as the source for the training prioritization and subsequent training development.
- task train prioritization for a task revision can refer to a previous task revision for its train prioritization
- task train prioritization can be evaluated using the task difficulty, importance, and frequency evaluation model
- the decision on task train prioritization can be iterated for a given task revision. For example, feedback on actual performance in the field can result in iterating the evaluation of the task/subtask train prioritization



ICN-S6000T-B6865-00043-001-01

Fig 1 UoF Task Train Prioritization

2.1 TaskTrainDecision

2.1.1 Class definition

TaskTrainDecision is a <<class>> that defines how a Task (refer to Chap 7.3.32) is to be managed from a training perspective.

2.1.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 taskTrainDecisionIterationIdentifier

taskTrainDecisionIterationIdentifier is an identifier that establishes a unique iteration for a TaskTrainDecision and to differentiate it from other iterations of TaskTrainDecision.

Valid identifier class values:

- ID (SX001G:taskTrainDecisionIterationIdentifier)

2.1.2.2 taskTrainDecisionRationale (zero, one or many)

taskTrainDecisionRationale is a description that gives more information about the justification for the choice on further training analysis for the Task. Refer to Chap 7.3.32.

2.1.2.3 taskTrainDecisionIterationRationale (zero, one or many)

taskTrainDecisionIterationRationale is a description that gives more information on the justification for iterating the decision on further training analysis for the Task. Refer to Chap 7.3.32.

2.1.2.4 taskTrainDecisionIterationDate (zero or one)

taskTrainDecisionIterationDate is a date that specifies when the decision on further training analysis for the Task (refer to Chap 7.3.32) was revised.

2.1.2.5 taskTrainDecisionIterationStatus (zero or one)

taskTrainDecisionIterationStatus is a state that identifies the maturity of the task train decision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)
- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

2.2 TaskTrainDecisionByEvaluation

2.2.1 Class definition

TaskTrainDecisionByEvaluation is a TaskTrainDecision where the characteristics of the Task are evaluated in order to decide its training level.

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of SubtaskTrainDecision

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- KnowledgeSkillAttitudeRequirementItem. Refer to Chap 7.3.33.

- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 taskTrainDecisionIterationIdentifier
Inherited from TaskTrainDecision.

2.2.2.2 taskTrainDecisionRationale (zero, one or many)
Inherited from TaskTrainDecision.

2.2.2.3 taskTrainDecisionIterationRationale (zero, one or many)
Inherited from TaskTrainDecision.

2.2.2.4 taskTrainDecisionIterationDate (zero or one)
Inherited from TaskTrainDecision.

2.2.2.5 taskTrainDecisionIterationStatus (zero or one)
Inherited from TaskTrainDecision.

2.2.2.6 taskTargetAudience
taskTargetAudience is a classification that identifies the kind of personnel that is to be trained to perform the Task. Refer to Chap 7.3.32.

Examples:

- operator
- product support

Valid values:

- O (SX001G:operatorTask)
- S (SX001G:productSupportTask)

2.2.2.7 taskHumanInteractionRequirement
taskHumanInteractionRequirement is a classification that identifies the type of human coordination that needs to take place when performing the Task. Refer to Chap 7.3.32.

Examples:

- collective
- individual

Valid values:

- C (SX001G:collectiveTask)
- I (SX001G:individualTask)

2.2.2.8 taskDifficultyCategory
taskDifficultyCategory is a classification that identifies the complexity of a Task (refer to Chap 7.3.32) from a human performance perspective.

Note

Task difficulty category can be derived from eg, the following task aspects:

- Personal safety

- Time it takes to perform the task
- Need for human interaction
- Conditions under which the task is to be performed

Valid values:

- `H` (SX001G:highTaskDifficulty)
- `L` (SX001G:lowTaskDifficulty)
- `M` (SX001G:mediumTaskDifficulty)

2.2.2.9 taskImportanceCategory

`taskImportanceCategory` is a classification that identifies possible adverse effects that the performance of the `Task` (refer to Chap 7.3.32) can have with respect to cost (damage) and availability for the item under analysis.

Note

Task importance category can be derived from eg, the following task aspects:

- Mission (operability) impact
- Product integrity impact
- Function essentiality

Valid values:

- `H` (SX001G:highImportanceTask)
- `L` (SX001G:lowImportanceTask)
- `M` (SX001G:mediumImportanceTask)

2.2.2.10 taskFrequencyCategory

`taskFrequencyCategory` is a classification that identifies how often the `Task` (refer to Chap 7.3.32) is to be performed.

Example:

- task frequency category is based on how many times a task (including similar tasks) is to be performed under a given time period (usually defined per year).

Valid values:

- `I` (SX001G:infrequentTask)
- `M` (SX001G:moderatlyFrequentTask)
- `V` (SX001G:veryFrequentTask)

2.2.2.11 taskTrainingLevel

`taskTrainingLevel` is a classification that identifies the priority for training for the performer to be able to carry out the `Task` (refer to Chap 7.3.32) within its defined task performance requirement.

Note

Training level is often referred to as training priority.

Valid values:

- `1` (SX001G:veryHighPriorityTrainingTask)
- `2` (SX001G:highPriorityTrainingTask)
- `3` (SX001G:moderatePriorityTrainingTask)
- `4` (SX001G:lowPriorityTrainingTask)

- 5 (SX001G:noFormalTrainingRequiredTask)

2.2.2.12 taskTrainingLevelRationale (zero, one or many)
taskTrainingLevelRationale is a description that gives more information on the justification for selecting the defined taskTrainingLevel.

2.2.2.13 taskTrainingDifficulty
taskTrainingDifficulty is a classification that identifies the complexity involved in training for the Task. Refer to Chap 7.3.32.

Valid values:

- H (SX001G:highDifficultyToTrainTask)
- L (SX001G:lowDifficultyToTrainTask)
- M (SX001G:moderateDifficultyToTrainTask)

2.2.2.14 taskTrainingLevelDecision
taskTrainingLevelDecision is a classification that identifies the final decision on task training level based on the task difficulty, importance, and frequency analysis as well as additional considerations such as task training difficulty, customer stipulations and regulations.

Valid values:

- 1 (SX001G:veryHighPriorityTrainingTask)
- 2 (SX001G:highPriorityTrainingTask)
- 3 (SX001G:moderatePriorityTrainingTask)
- 4 (SX001G:lowPriorityTrainingTask)
- 5 (SX001G:noFormalTrainingRequiredTask)

2.2.2.15 taskTrainingLevelDecisionRationale (zero, one or many)
taskTrainingLevelDecisionRationale is a description that gives more information on the justification for the final taskTrainingLevelDecision.

2.3 TaskTrainDecisionByPreviousTaskRevision

2.3.1 Class definition

TaskTrainDecisionByPreviousTaskRevision is a TaskTrainDecision that refers to a previous revision of the Task (refer to Chap 7.3.32) as the source for its training analysis.

2.3.1.1 Implementations
This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 taskTrainDecisionIterationIdentifier
Inherited from TaskTrainDecision.

2.3.2.2 taskTrainDecisionRationale (zero, one or many)
Inherited from TaskTrainDecision.

2.3.2.3 taskTrainDecisionIterationRationale (zero, one or many)
Inherited from TaskTrainDecision.

2.3.2.4 taskTrainDecisionIterationDate (zero or one)
Inherited from TaskTrainDecision.

2.3.2.5 taskTrainDecisionIterationStatus (zero or one)
Inherited from TaskTrainDecision.

2.4 TaskTrainDecisionByTaskGrouping

TaskTrainDecisionByTaskGrouping is a TaskTrainDecision that refers to another Task (refer to Chap 7.3.32) as the source for its training analysis.

2.4.1.1 Associations
This class has the following associations:

- A directed definedBy association with one instance of Task

2.4.1.2 Implementations
This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 taskTrainDecisionIterationIdentifier
Inherited from TaskTrainDecision.

2.4.2.2 taskTrainDecisionRationale (zero, one or many)
Inherited from TaskTrainDecision.

2.4.2.3 taskTrainDecisionIterationRationale (zero, one or many)
Inherited from TaskTrainDecision.

2.4.2.4 taskTrainDecisionIterationDate (zero or one)
Inherited from TaskTrainDecision.

2.4.2.5 taskTrainDecisionIterationStatus (zero or one)
Inherited from TaskTrainDecision.

Chapter 7.3.38

Units of functionality - UoF Training Analysis and Design Message Content

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.1	Units of functionality - UoF Aggregated Element
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.5	Units of functionality - UoF Circuit Breaker
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.12	Units of functionality - UoF Environment Definition
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.15	Units of functionality - UoF Learning Assessment

Chap No./Document No.	Title
Chap 7.3.16	Units of functionality - UoF Learning Gap
Chap 7.3.20	Units of functionality - UoF Organization
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.24	Units of functionality - UoF Product Usage Context
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.28	Units of functionality - UoF Security Classification
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity
Chap 7.3.43	Units of functionality - UoF Training Method
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Training Analysis and Design Message Content UoF defines the collection of information that can be exchanged for training analysis and design activities in the context of the S-Series IPS specifications.

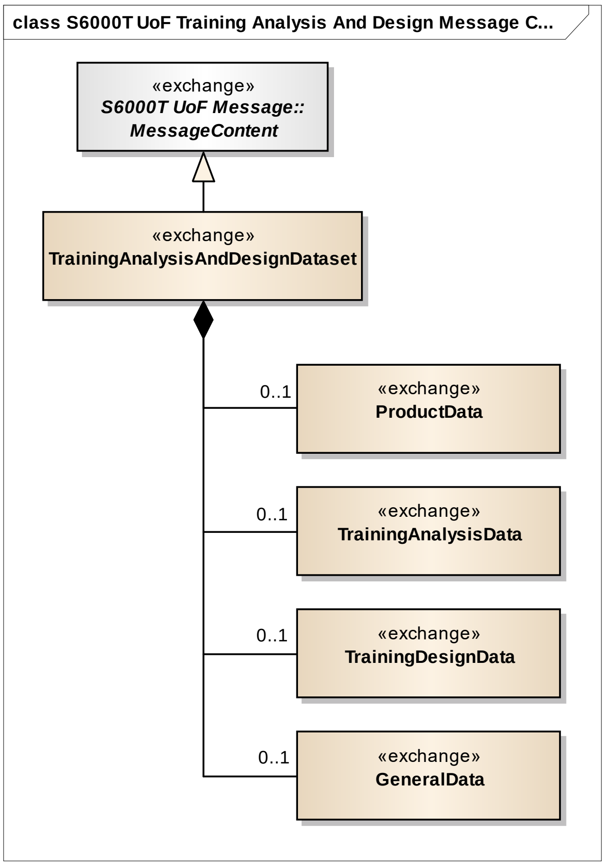
2 UoF Training Analysis and Design Message Content

The key feature of the UoF Training Analysis and Design Message Content data model (refer to Fig 1 thru Fig 5) is that it organizes data that can exchanged into four major subsections:

- Product structure and parts data that mainly originates from support analysis activities which precedes training analysis and design. Refer to Fig 2.
- Task, job, and competency data which is the basis for the training analysis. Refer to Fig 3.
- Data identified during training design. Refer to Fig 4.
- General data which allows for further characterization of the data that has been defined during training analysis and design. Refer to Fig 5.

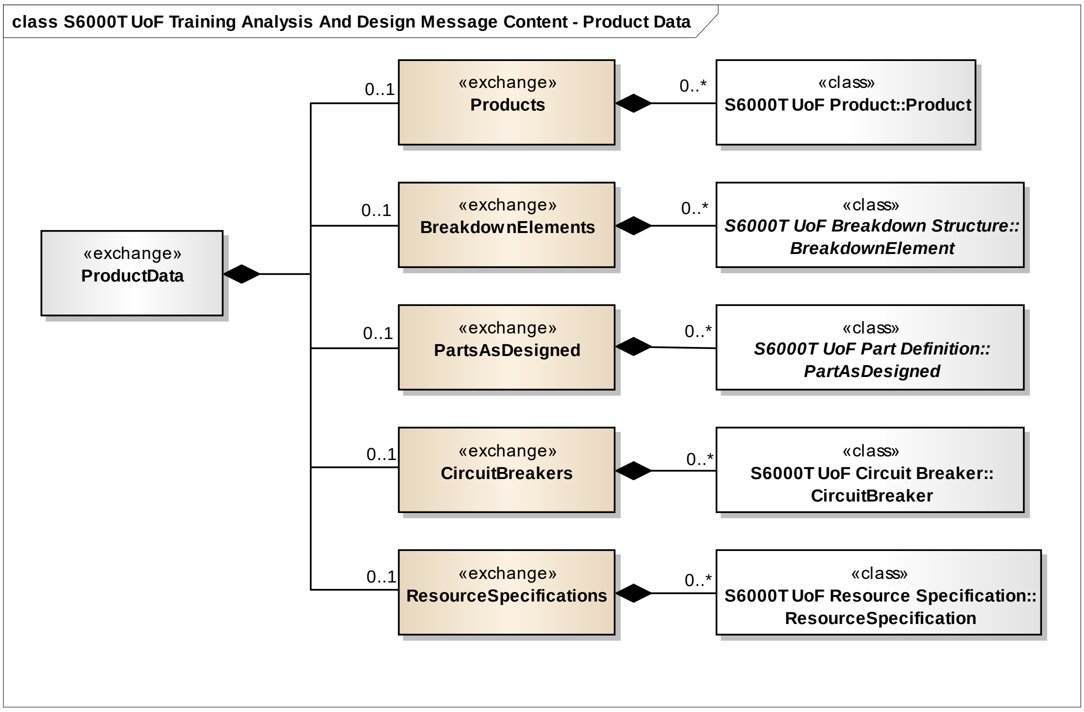
Note

Each branch in the exchange definition often encapsulates additional data which are not explicitly illustrated in the respective figure. For more details refer to the respective UoF where the included classes are defined or further characterized.



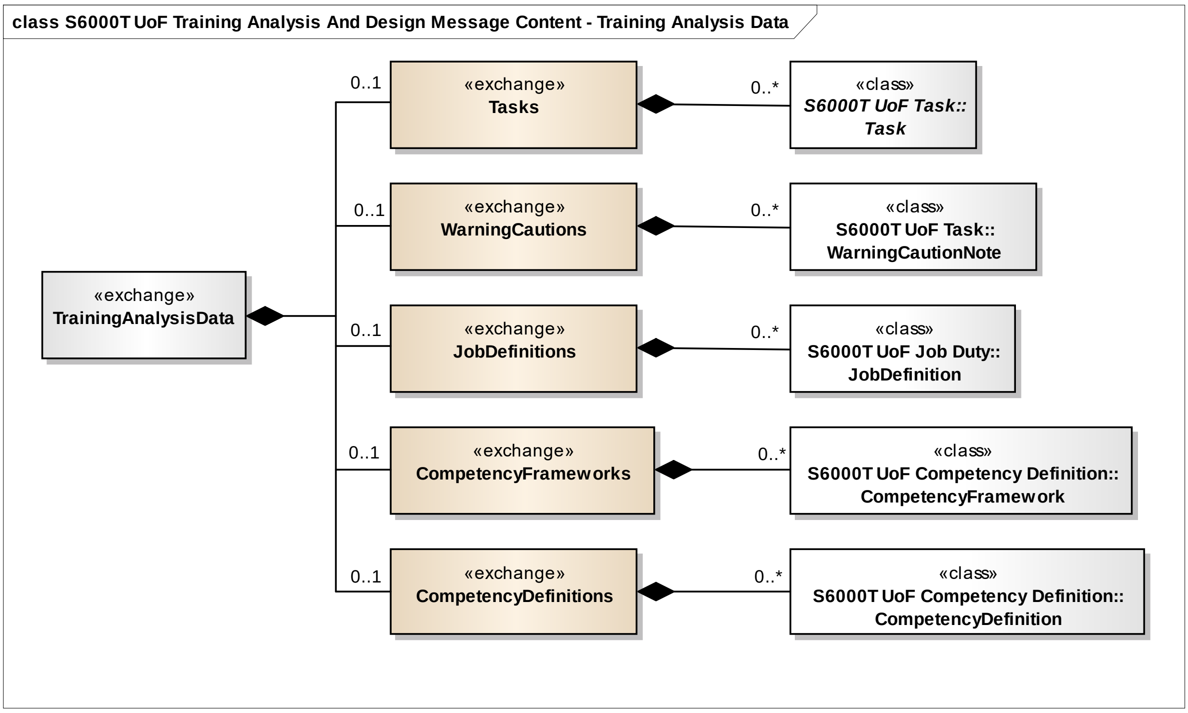
ICN-S6000T-B6865-00068-001-01

Fig 1 S6000T UoF Training Analysis and Design Message Content



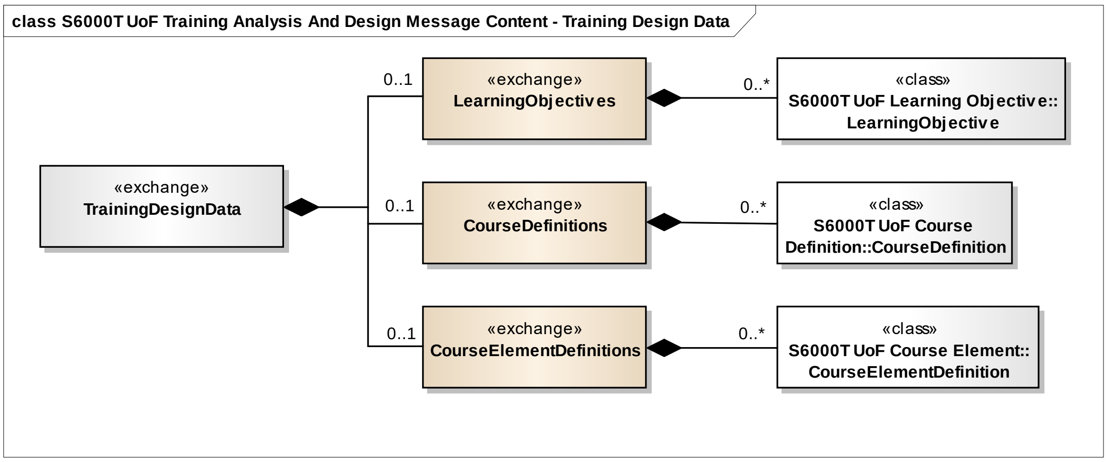
ICN-S6000T-B6865-00069-002-01

Fig 2 S6000T UoF Training Analysis and Design Message Content - Product Data



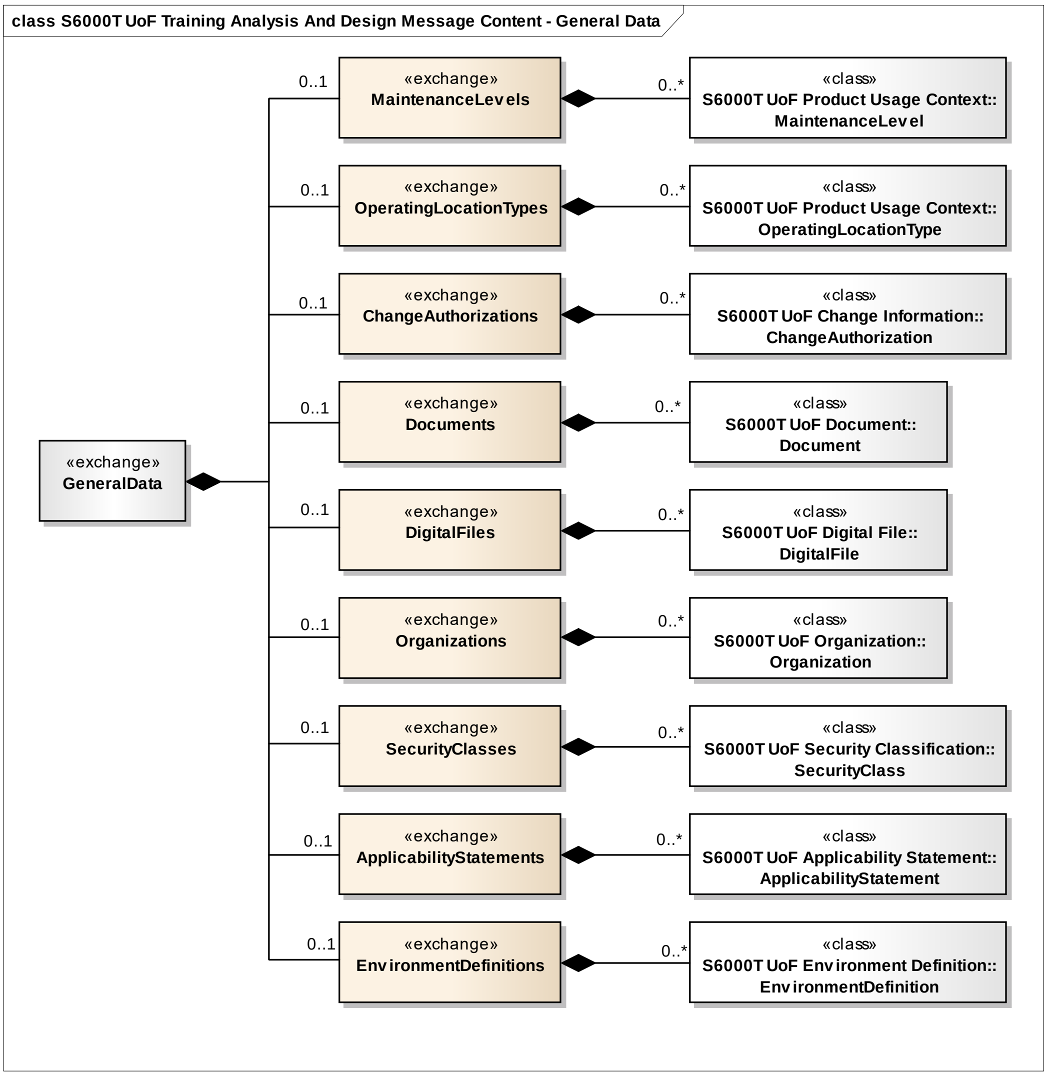
IC ICN-S6000T-B6865-00070-002-01

Fig 3 S6000T UoF Training Analysis and Design Message Content - Training Analysis Data



ICN-S6000T-B6865-00071-002-01

Fig 4 S6000T UoF Training Analysis and Design Message Content - Training Design Data



ICN-S6000T-B6865-00072-002-01

Fig 5 S6000T UoF Training Analysis and Design Message Content - General Data

2.1 ApplicabilityStatements

2.1.1 Class definition

`ApplicabilityStatements` is a wrapper element that contains instances of `ApplicabilityStatement` (refer to Chap 7.3.1) in scope for a data exchange.

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `ApplicabilityStatement`. Refer to Chap 7.3.2.

2.2 BreakdownElements

2.2.1 Class definition

`BreakdownElements` is a wrapper element that contains instances of `BreakdownElement` (refer to Chap 7.3.3) in scope for a data exchange.

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of BreakdownElement (refer to Chap 7.3.3)

2.3 ChangeAuthorizations

2.3.1 Class definition

ChangeAuthorizations is a wrapper element that contains instances of ChangeAuthorization (refer to Chap 7.3.4) in scope for a data exchange.

2.3.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of ChangeAuthorization. Refer to Chap 7.3.4.

2.4 CircuitBreakers

2.4.1 Class definition

CircuitBreakers is a wrapper element that contains instances of CircuitBreaker (refer to Chap 7.3.5) in scope for a data exchange.

2.4.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of CircuitBreaker. Refer to Chap 7.3.5.

2.5 CompetencyDefinitions

2.5.1 Class definition

CompetencyDefinitions is a wrapper element that contains instances of CompetencyDefinition (refer to Chap 7.3.6) in scope for a data exchange.

2.5.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of CompetencyDefinition. Refer to Chap 7.3.6.

2.6 CompetencyFrameworks

2.6.1 Class definition

CompetencyFrameworks is a wrapper element that contains instances of CompetencyFramework (refer to Chap 7.3.6) in scope for a data exchange.

2.6.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of CompetencyFramework. Refer to Chap 7.3.6.

2.7 CourseDefinitions

2.7.1 Class definition

CourseDefinitions is a wrapper element that contains instances of CourseDefinition (refer to Chap 7.3.7) in scope for a data exchange.

2.7.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of CourseDefinition (refer to Chap 7.3.7)

2.8 CourseElementDefinitions

2.8.1 Class definition

CourseElementDefinitions is a wrapper element that contains instances of CourseElementDefinition (refer to Chap 7.3.8) in scope for a data exchange.

2.8.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of CourseElementDefinition. Refer to Chap 7.3.8.

2.9 DigitalFiles

2.9.1 Class definition

DigitalFiles is a wrapper element that contains instances of DigitalFile (refer to Chap 7.3.10) in scope for a data exchange.

2.9.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of DigitalFile. Refer to Chap 7.3.10.

2.10 Documents

2.10.1 Class definition

Documents is a wrapper element that contains instances of Document (refer to Chap 7.3.11) in scope for a data exchange.

2.10.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of Document. Refer to Chap 7.3.11.

2.11 EnvironmentDefinitions

2.11.1 Class definition

EnvironmentDefinitions is a wrapper element that contains instances of EnvironmentDefinition (refer to Chap 7.3.12) in scope for a data exchange.

2.11.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of EnvironmentDefinition. Refer to Chap 7.3.12.

2.12 GeneralData

2.12.1 Class definition

GeneralData defines a subset of the TrainingAnalysisAndDesignDataset <<exchange>> and includes data that can assist in the characterization of data which has been defined during training analysis and design.

2.12.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of ApplicabilityStatements
- An aggregate association with zero or one instance of ChangeAuthorizations
- An aggregate association with zero or one instance of DigitalFiles
- An aggregate association with zero or one instance of Documents
- An aggregate association with zero or one instance of EnvironmentDefinitions
- An aggregate association with zero or one instance of MaintenanceLevels
- An aggregate association with zero or one instance of OperatingLocationTypes
- An aggregate association with zero or one instance of Organizations
- An aggregate association with zero or one instance of SecurityClasses

2.13 JobDefinitions

2.13.1 Class definition

JobDefinitions is a wrapper element that contains instances of JobDefinition (refer to Chap 7.3.14) in scope for a data exchange.

Note

Job definition also embeds information on:

- Duty definitions and duty tasks
- Training entry requirements. Refer to Chap 7.3.39.
- Occupational backgrounds. Refer to Chap 7.3.31.
- Knowledge skills and attitude gaps. Refer to Chap 7.3.16.
- Curriculums. Refer to Chap 7.3.9.

2.13.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of JobDefinition. Refer to Chap 7.3.14.

2.14 LearningObjectives

2.14.1 Class definition

LearningObjectives is a wrapper element that contains instances of LearningObjective (refer to Chap 7.3.17) in scope for a data exchange.

Note

Learning objective also embeds information on:

- Learning assessment. Refer to Chap 7.3.15.
- Training media and fidelity. Refer to Chap 7.3.41.
- Training methods. Refer to Chap 7.3.43.

2.14.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of LearningObjective. Refer to Chap 7.3.17.

2.15 MaintenanceLevels

2.15.1 Class definition

MaintenanceLevels is a wrapper element that contains instances of MaintenanceLevel (refer to Chap 7.3.24) in scope for a data exchange.

2.15.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of MaintenanceLevel. Refer to Chap 7.3.24.

2.16 OperatingLocationTypes

2.16.1 Class definition

OperatingLocationTypes is a wrapper element that contains instances of OperatingLocationType (refer to Chap 7.3.24) in scope for a data exchange.

2.16.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of OperatingLocationType. Refer to Chap 7.3.24.

2.17 Organizations

2.17.1 Class definition

Organizations is a wrapper element that contains instances of Organization (refer to Chap 7.3.20) in scope for a data exchange.

2.17.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of Organization. Refer to Chap 7.3.20.

2.18 PartsAsDesigned

2.18.1 Class definition

PartsAsDesigned is a wrapper element that contains all instances of PartAsDesigned (refer to Chap 7.3.21) in scope for a data exchange.

2.18.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of PartAsDesigned. Refer to Chap 7.3.21.

2.19 ProductData

2.19.1 Class definition

ProductData defines a subset of the TrainingAnalysisAndDesignDataset <<exchange>> and includes product structure and part data relevant to training analysis and design.

2.19.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of BreakdownElements
- An aggregate association with zero or one instance of CircuitBreakers
- An aggregate association with zero or one instance of PartsAsDesigned
- An aggregate association with zero or one instance of Products
- An aggregate association with zero or one instance of ResourceSpecifications

2.20 Products

2.20.1 Class definition

`Products` is a wrapper element that contains all instances of `Product` (refer to Chap 7.3.23) in scope for a data exchange.

2.20.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `Product`. Refer to Chap 7.3.23.

2.21 ResourceSpecifications

2.21.1 Class definition

`ResourceSpecifications` is a wrapper element that contains all instances of `ResourceSpecification` (refer to Chap 7.3.27) in scope for a data exchange.

2.21.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `ResourceSpecification`. Refer to Chap 7.3.27.

2.22 SecurityClasses

2.22.1 Class definition

`SecurityClasses` is a wrapper element that contains all instances of `SecurityClass` (refer to Chap 7.3.28) in scope for a data exchange.

2.22.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `SecurityClass`. Refer to Chap 7.3.28.

2.23 Tasks

2.23.1 Class definition

`Tasks` is a wrapper element that contains all instances of `Task` (refer to Chap 7.3.32) in scope for a data exchange.

Note

Task also embeds information on:

- Task performance objective. Refer to Chap 7.3.24.
- Task and subtask train prioritization. Refer to Chap 7.3.37 and Chap 7.3.30.
- Task knowledge, skills, and attitude requirements. Refer to Chap 7.3.33.

2.23.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `Task` Refer to Chap 7.3.32.

2.24 TrainingAnalysisAndDesignDataset

2.24.1 Class definition

`TrainingAnalysisAndDesignDataset` is an `<<exchange>>` definition that includes information from training analysis and design which is of interest to share between different project shareholders.

2.24.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of GeneralData
- An aggregate association with zero or one instance of ProductData
- An aggregate association with zero or one instance of TrainingAnalysisData
- An aggregate association with zero or one instance of TrainingDesignData

2.25 TrainingAnalysisData

2.25.1 Class definition

TrainingAnalysisData defines a subset of the TrainingAnalysisAndDesignDataset <<exchange>> and includes data that is defined during training analysis activities, maintaining the thread back to the Product and Task data as defined during Product design and Logistics Support Analysis (LSA).

2.25.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of CompetencyDefinitions
- An aggregate association with zero or one instance of CompetencyFrameworks
- An aggregate association with zero or one instance of JobDefinitions
- An aggregate association with zero or one instance of Tasks
- An aggregate association with zero or one instance of WarningCautions

2.26 TrainingDesignData

2.26.1 Class definition

TrainingDesignData defines a subset of the TrainingAnalysisAndDesignDataset <<exchange>> and includes data defined during training design activities, maintaining the thread back to training analysis.

2.26.1.1 Associations

This class has the following associations:

- An aggregate association with zero or one instance of CourseDefinitions
- An aggregate association with zero or one instance of CourseElementDefinitions
- An aggregate association with zero or one instance of LearningObjectives

2.27 WarningCautions

2.27.1 Class definition

WarningCautions is a wrapper element that contains all instances of WarningCautionNote (refer to Chap 7.3.32) in scope for a data exchange.

2.27.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of WarningCautionNote. Refer to Chap 7.3.32.

Chapter 7.3.39

Units of functionality - UoF Training Entry Requirement

Table of contents

Page

List of tables

List of figures

References

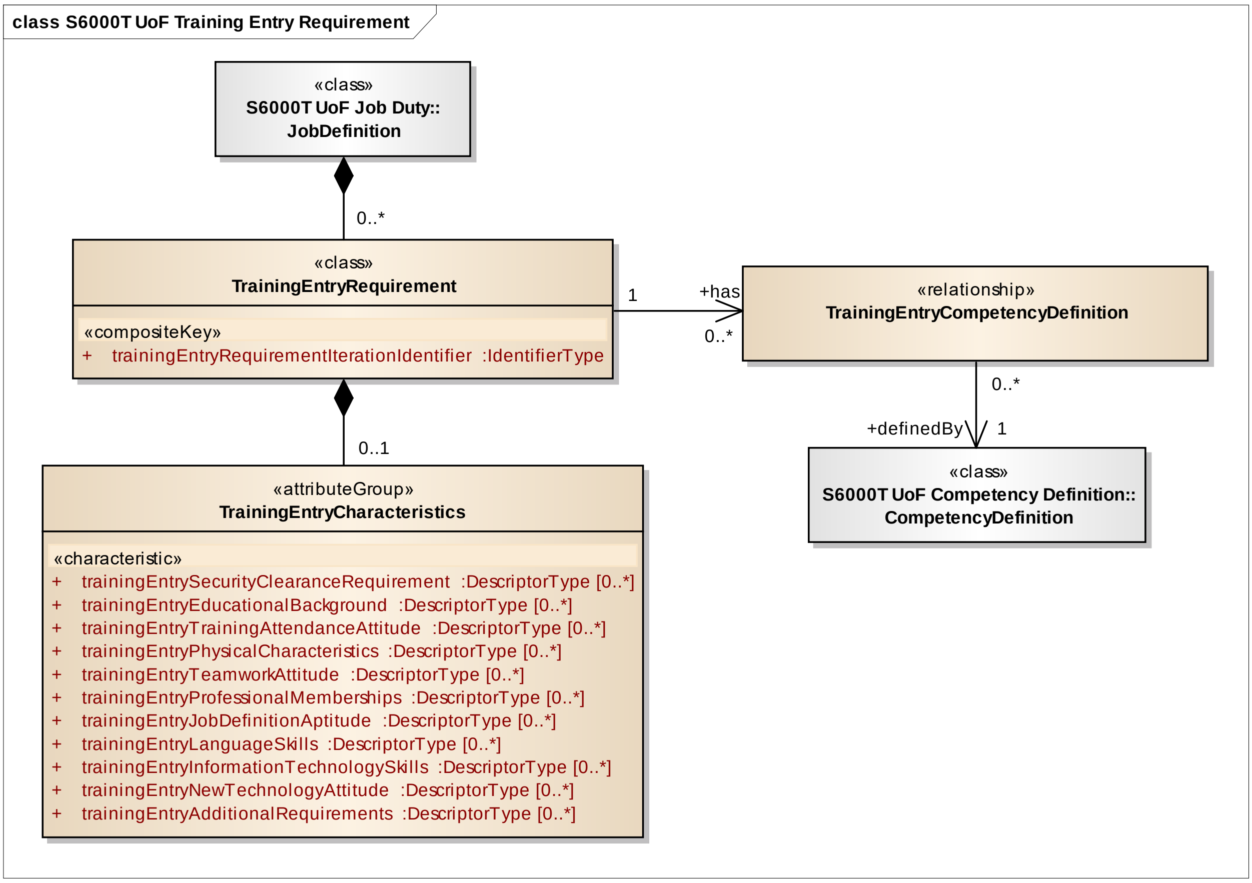
Table 1 References

Chap No./Document No.	Title
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Training Entry Requirement UoF provides the capability to define criteria which must be fulfilled before a person can enter the defined training program for a particular job.

2 UoF Training Entry Requirement



ICN-S6000T-B6865-00098-001-01

Fig 1 UoF Training Entry Requirement

2.1 TrainingEntryCharacteristics

2.1.1 Attribute group definition

`TrainingEntryCharacteristics` is an `<<attributeGroup>>` that collects training audience intrinsic entry requirements.

2.1.2 Attributes

2.1.2.1 trainingEntrySecurityClearanceRequirement (zero, one or many)

`trainingEntrySecurityClearanceRequirement` is a description that provides information about the level of security clearance required of a trainee to participate in a curriculum, course, or course element.

2.1.2.2 trainingEntryEducationalBackground (zero, one or many)

`trainingEntryEducationalBackground` is a description that provides information about the schools, areas of study, and degrees or certificates required of a trainee prior to starting a curriculum, course or course element.

2.1.2.3 trainingEntryTrainingAttendanceAttitude (zero, one or many)

`trainingEntryTrainingAttendanceAttitude` is a description that provides information about the emotions, beliefs, and behaviors required of a trainee regarding participation in formal training activities prior to starting a curriculum, course, or course element.

2.1.2.4 trainingEntryPhysicalCharacteristics (zero, one or many)
trainingEntryPhysicalCharacteristics is a description that provides information about the bodily strength, stamina, coordination, balance, and flexibility required of a trainee prior to starting a curriculum, course, or course element.

2.1.2.5 trainingEntryTeamworkAttitude (zero, one or many)
trainingEntryTeamworkAttitude is a description that provides information about the emotions, beliefs, and behaviors required of a trainee regarding working with others versus individually prior to starting a curriculum, course, or course element.

2.1.2.6 trainingEntryProfessionalMemberships (zero, one or many)
trainingEntryProfessionalMemberships is a description that provides information about the organizations or affiliations for a particular job or profession to which a trainee must belong prior to starting a curriculum, course, or course element.

2.1.2.7 trainingEntryJobDefinitionAptitude (zero, one or many)
trainingEntryJobDefinitionAptitude is a description that provides information about natural talents or special abilities for a job which are required of a trainee prior to starting a curriculum, course, or course element.

2.1.2.8 trainingEntryLanguageSkills (zero, one or many)
trainingEntryLanguageSkills is a description that provides information about the language and proficiency level in that language that are required of a trainee prior to starting a curriculum, course, or course element.

2.1.2.9 trainingEntryInformationTechnologySkills (zero, one or many)
trainingEntryInformationTechnologySkills is a description that provides information about the ability required of a trainee to use digital systems, equipment, and software prior to starting a curriculum, course, or course element.

2.1.2.10 trainingEntryNewTechnologyAttitude (zero, one or many)
trainingEntryNewTechnologyAttitude is a description that provides information about the emotions, beliefs, and behaviors required of a trainee regarding the use of digital systems, equipment, and software that is new to them prior to starting a curriculum, course, or course element.

2.1.2.11 trainingEntryAdditionalRequirements (zero, one or many)
trainingEntryAdditionalRequirements is a description that provides other details about what is required of a trainee prior to starting a curriculum, course, or course element that is in addition to information described by other TrainingEntryCharacteristics.

2.2 TrainingEntryCompetencyDefinition

2.2.1 Class definition

TrainingEntryCompetencyDefinition is a <<relationship>> where an instance of TrainingEntryRequirement refers to a defined CompetencyDefinition (refer to Chap 7.3.6) as the entry requirement.

2.2.1.1 Associations
This class has the following associations:

- A directed definedBy association with one instance of CompetencyDefinition. Refer to Chap 7.3.6.

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3 TrainingEntryRequirement

2.3.1 Class definition

TrainingEntryRequirement is a <<class>> that defines prerequisites that must be fulfilled by a person that is to join a training program for a job.

2.3.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TrainingEntryCharacteristics
- A directed has association with zero, one or many instances of TrainingEntryCompetencyDefinition

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 trainingEntryRequirementIterationIdentifier

trainingEntryRequirementIterationIdentifier is an identifier that establishes a unique designator for a TrainingEntryRequirement and to differentiate it from other instances of TrainingEntryRequirement.

Valid identifier class values:

- ID (SX001G:trainingEntryRequirementIterationIdentifier)

Chapter 7.3.40

Units of functionality - UoF Training Gap

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.9	Units of functionality - UoF Curriculum and Course Plan
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.30	Units of functionality - UoF Subtask Train Prioritization
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
Chap 7.3.37	Units of functionality - UoF Task Train Prioritization
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

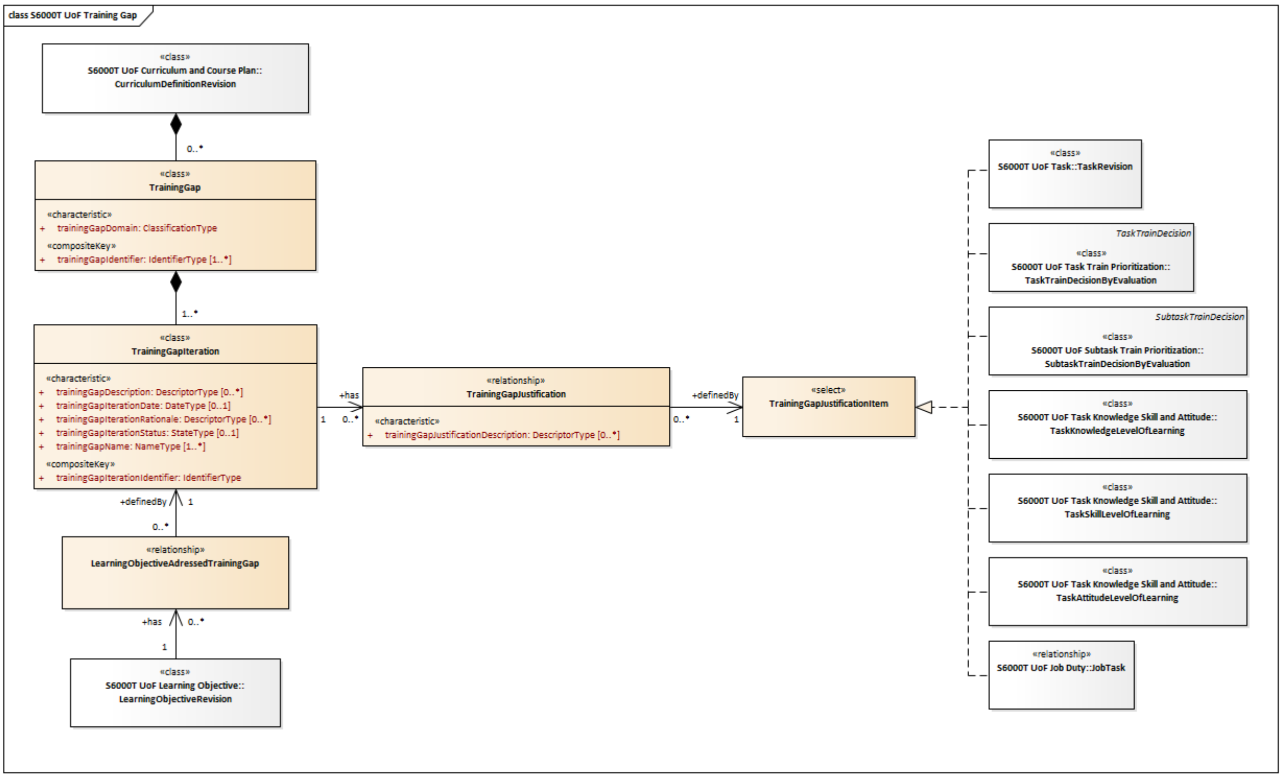
1 General

The Training Gap UoF provides the capability to define gaps between the existing curriculum and the current formal training solution that is required to perform new or changed tasks.

2 UoF Training Gap

Key features of the UoF Training Gap data model (refer to Fig 1) are:

- The definition of `TrainigGap` and its characteristic can be iterated using the `TrainingGapIteration`
- A `TrainigGapIteration` can be associated with a `TrainigGapJustification` to which the training gap is related
- A `TrainigGapIteration` can be addressed by one or many instances of `LearningObjective` via `LearningObjectiveAdressTrainigGap`



ICN-S6000T-B6865-00105-001-01

Fig 1 UoF Training Gap

2.1 TrainingGap

2.1.1 Class definition

`TrainingGap` is a `<<class>>` that defines the gap between the behavior, conditions, and standards, required to perform a new or changed task, and any pre-existing learning objectives included in a training curriculum.

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `TrainingGapIteration`

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 trainingGapDefinitionIdentifier (one or many)

trainingGapIdentifier is an identifier that establishes a unique designator for a TrainingGap differentiates it from other instances of TrainingGap.

2.1.2.2 trainingGapDefinitionDomain

trainingGapDomain is a classification that indicates the learning domain to which the gap is written.

2.2 TrainigGapIteration

2.2.1 Class definition

TrainingGapIteration is a <<class>> representing an iteration applied to a TrainingGap.

2.2.1.1 Associations

This class has the following associations:

- A directed has association with zero, one or many instances of TrainingGapJustification

2.2.1.2 Class members

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 trainingGapIterationIdentifier

trainingGapIterationIdentifier is an identifier that establishes a unique designator for a TrainingGapIteration and differentiates it from other instances of TrainingGapIteration.

2.2.2.2 trainingGapName (one or many)

trainingGapName is a name by which the TrainingGap is known and can be easily referenced.

2.2.2.3 trainingGapDescription (zero, one or many)

trainingGapDescription is a description that gives more information on the identified TrainingGap.

2.2.2.4 trainingGapIterationRationale (zero, one or many)
trainingGapIterationRationale is a description that gives more information about the justification for revising the TrainingGap.

trainingGapIterationDate (zero or one)

trainingGapIterationDate is a date that specifies when the TrainingGap was revised.

trainingGapIterationStatus (zero or one)

trainingGapIterationStatus is a state that identifies the maturity of the TrainingGapIteration.

TrainingGapJustification

2.2.3 Relationship definition

TrainingGapJustification is a <<relationship>> that identifies the source from which the need for additional training has been identified.

2.2.3.1 Associations
This class has the following associations:

- A directed definedBy association with one instance of TrainingGapJustificationItem

2.2.3.2 Implementations
This class has the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.4 Attributes

2.2.4.1 trainingGapJustificationDescription (zero, one or many)
trainingGapJustificationDescription is a description that gives more information on why there is a training gap in the context of the TrainingGapJustificationItem.

2.3 TrainingGapJustificationItem

2.3.1 Interface definition

TrainingGapJustificationItem is a <<select>> interface that identifies items on which can require additional training.

Class member

This <<select>> interface includes the following class members:

- TaskRevision. Refer to Chap 7.3.32.
- TaskTrainDecisionByEvaluation. Refer to Chap 7.3.37.
- SubtaskTrainDecisionByEvaluation. Refer to Chap 7.3.30.
- TaskKnowledgeLevelOfLearning. Refer to Chap 7.3.33.
- TaskSkillLevelOfLearning. Refer to Chap 7.3.33.

- TaskAttitudeLevelOfLearning. Refer to Chap 7.3.33.
- JobTask. Refer to Chap 7.3.14.

2.4 LearningObjectiveAdressedTrainingGap

2.4.1 Relationship definition

LearningObjectiveAddressedTrainingGap is a <<relationship>> that identifies a training gap which the associated LearningObjectiveRevision addresses.

2.4.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of TrainingGapIteration

2.4.1.2 Implementations

This class has the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

Chapter 7.3.41

Units of functionality - UoF Training Media and Fidelity

Table of contents

Page

List of tables

List of figures

References

Table 1 References

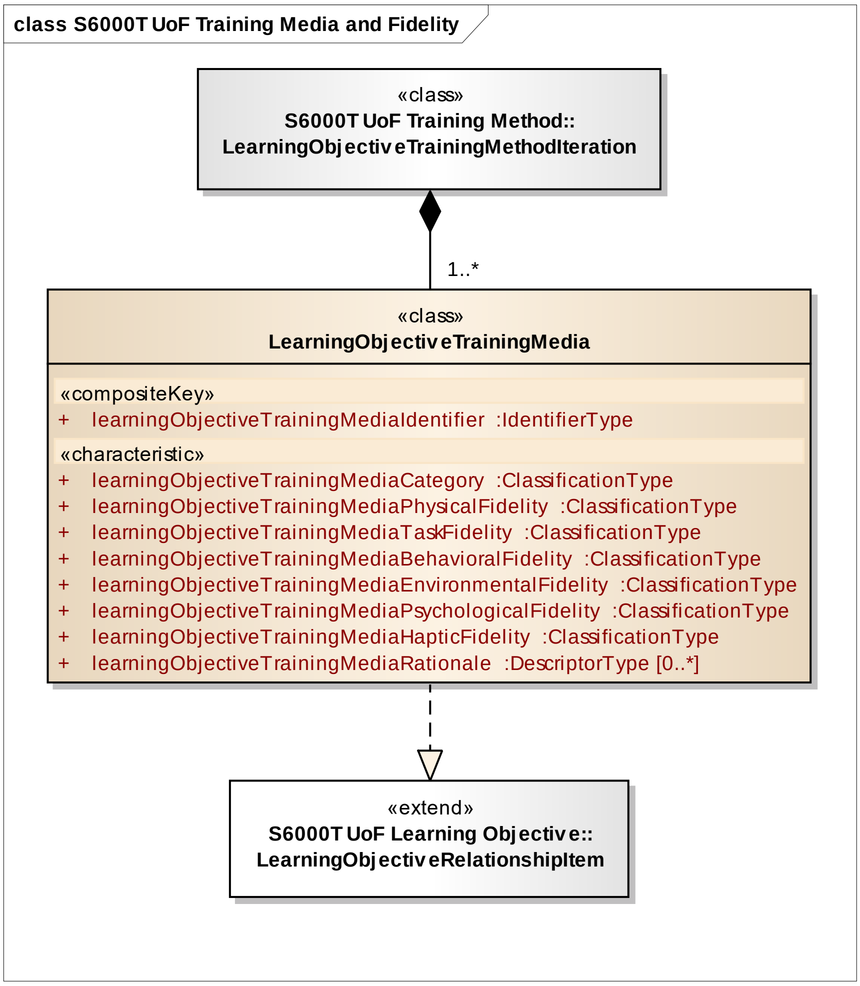
Chap No./Document No.	Title
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.42	Units of functionality - UoF Training Media Resource
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Training Media and Fidelity UoF provides the capability to document ways to deliver learning content and determine its realism.

2 UoF Training Media and Fidelity

A key feature of the UoF Training Media and Fidelity data model, (refer to Fig 1), is that training media for a learning objective is always defined in the context of its defined training method.



ICN-S6000T-B6865-00101-001-01

Fig 1 UoF Training Media and Fidelity

2.1 LearningObjectiveTrainingMedia

2.1.1 Class definition

`LearningObjectiveTrainingMedia` is a `<<class>>` that identifies the means, form, or vehicle by which the `LearningObjective` (refer to Chap 7.3.17) instruction is formatted, stored, and delivered to the trainee.

2.1.1.1 Associations

This class has the following associations:

– A directed `has` association with zero, one or many instances of `TrainingMediaResource`. Refer to Chap 7.3.42.

2.1.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

– `DigitalFileReferencingItem`. Refer to Chap 7.3.10.

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- LearningObjectiveRelationshipItem. Refer to Chap 7.3.17.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 learningObjectiveTrainingMediaIdentifier

learningObjectiveTrainingMediaIdentifier is an identifier that establishes a unique designator for a LearningObjectiveTrainingMedia and to differentiate it from other instances of LearningObjectiveTrainingMedia.

Valid identifier class values:

- ID (SX001G:learningObjectiveTrainingMediaIdentifier)

2.1.2.2 learningObjectiveTrainingMediaCategory

learningObjectiveTrainingMediaCategory is a classification that identifies the delivery format of learning content to trainees.

Valid values:

- DIGI (SX001G:digitalTrainingMedia)
- ELEC (SX001G:electronicTrainingMedia)
- PRIN (SX001G:printedTrainingMedia)
- PHYS (SX001G:physicalTrainingMedia)

2.1.2.3 learningObjectiveTrainingMediaPhysicalFidelity

learningObjectiveTrainingMediaPhysicalFidelity is a classification that defines how closely a training intervention for the LearningObjective (refer to Chap 7.3.17) must imitate or replicate a person's real-world passive perception of the Product (refer to Chap 7.3.23) through the senses of sight, hearing, smell, touch, or taste.

Valid values:

- L (SX001G:lowTrainingMediaPhysicalFidelity)
- M (SX001G:mediumTrainingMediaPhysicalFidelity)
- H (SX001G:highTrainingMediaPhysicalFidelity)

2.1.2.4 learningObjectiveTrainingMediaTaskFidelity

learningObjectiveTrainingMediaTaskFidelity is a classification that defines how closely a training intervention for the LearningObjective (refer to Chap 7.3.17) must imitate or replicate the actual actions performed within a Task. Refer to Chap 7.3.32.

Valid values:

- L (SX001G:lowTrainingMediaTaskFidelity)
- M (SX001G:mediumTrainingMediaTaskFidelity)
- H (SX001G:highTrainingMediaTaskFidelity)

2.1.2.5 learningObjectiveTrainingMediaBehavioralFidelity

`learningObjectiveTrainingMediaBehavioralFidelity` is a classification that defines how closely a training intervention for the `LearningObjective` (refer to Chap 7.3.17) must imitate or replicate the real-world characteristics of the `Product`. Refer to Chap 7.3.23.

Valid values:

- `L` (SX001G:lowTrainingMediaBehavioralFidelity)
- `M` (SX001G:mediumTrainingMediaBehavioralFidelity)
- `H` (SX001G:highTrainingMediaBehavioralFidelity)

2.1.2.6 learningObjectiveTrainingMediaEnvironmentalFidelity

`learningObjectiveTrainingMediaEnvironmentalFidelity` is a classification that defines how closely a training intervention for the `LearningObjective` (refer to Chap 7.3.17) must imitate or replicate the real-world circumstances, events and/or conditions, under which the `Product` (refer to Chap 7.3.23) is used.

Valid values:

- `L` (SX001G:lowTrainingMediaEnvironmentalFidelity)
- `M` (SX001G:mediumTrainingMediaEnvironmentalFidelity)
- `H` (SX001G:highTrainingMediaEnvironmentalFidelity)

2.1.2.7 learningObjectiveTrainingMediaPsychologicalFidelity

`learningObjectiveTrainingMediaPsychologicalFidelity` is a classification that defines how closely a training intervention for the `LearningObjective` (refer to Chap 7.3.17) must imitate or replicate a person's real-world mental or emotional response to the conditions under which the `Product` (refer to Chap 7.3.23) will be used.

Valid values:

- `L` (SX001G:lowTrainingMediaPsychologicalFidelity)
- `M` (SX001G:mediumTrainingMediaPsychologicalFidelity)
- `H` (SX001G:highTrainingMediaPsychologicalFidelity)

2.1.2.8 learningObjectiveTrainingMediaHapticFidelity

`learningObjectiveTrainingMediaHapticFidelity` is a classification that defines how closely a training intervention for the `LearningObjective` (refer to Chap 7.3.17) must imitate or replicate a person's real-world active perception of the `Product` (refer to Chap 7.3.23) through the sense of touch or bodily interaction.

Valid values:

- `L` (SX001G:lowTrainingMediaHapticFidelity)
- `M` (SX001G:mediumTrainingMediaHapticFidelity)
- `H` (SX001G:highTrainingMediaHapticFidelity)

2.1.2.9 learningObjectiveTrainingMediaRationale (zero, one or many)

`learningObjectiveTrainingMediaRationale` is a description that gives more information on the justification for selecting the `learningObjectiveTrainingMediaCategory`.

Chapter 7.3.42

Units of functionality - UoF Training Media Resource

Table of contents

Page

List of tables

List of figures

References

Table 1 References

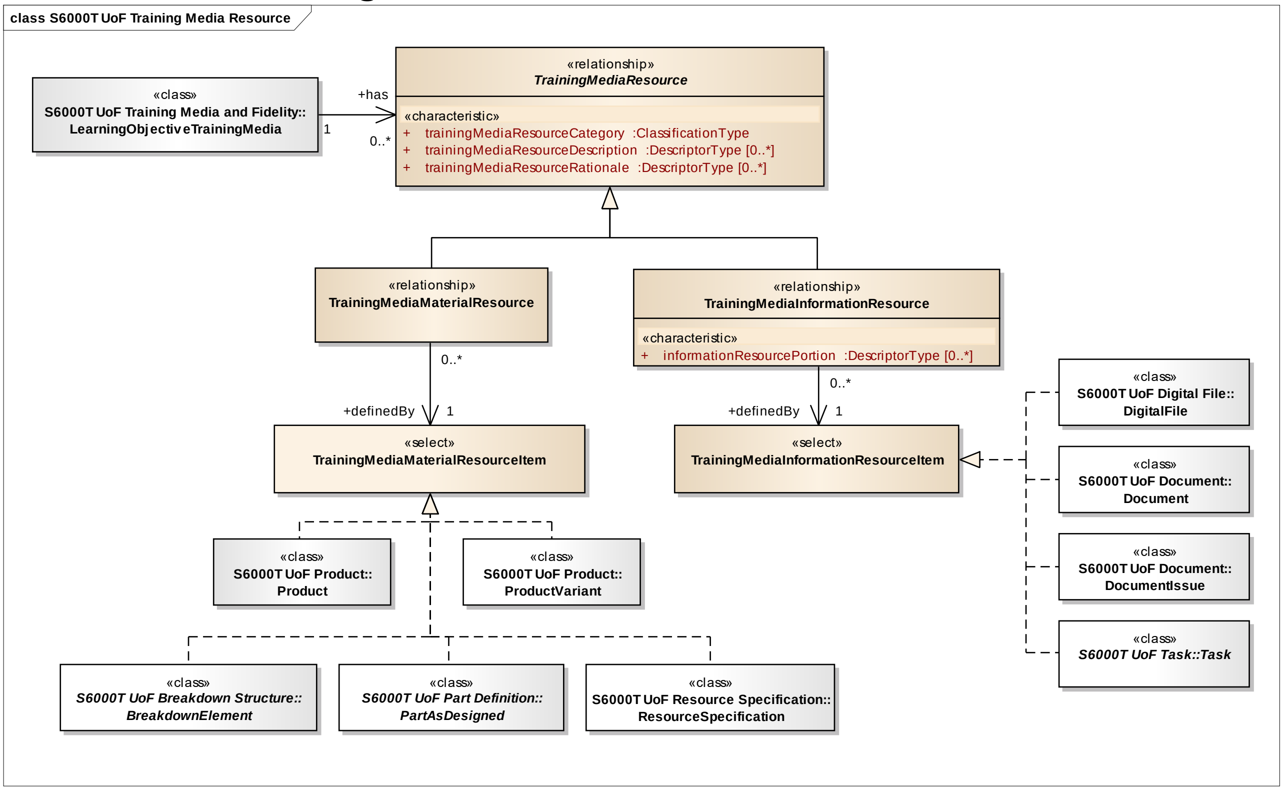
Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.10	Units of functionality - UoF Digital File
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark

Chap No./Document No.	Title
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.32	Units of functionality - UoF Task
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

The Training Media Resource UoF supports the identification resources which can or must be considered during training development.

2 UoF Training Media Resource



ICN-S6000T-B6865 00102-001-01

Fig 1 UoF Training Media Resource

2.1 TrainingMediaInformationResource

2.1.1 Class definition

`TrainingMediaInformationResource` is a `TrainingMediaResource` that exists in the form of information.

2.1.1.1 Associations

This class has the following associations:

– A directed `definedBy` association with one instance of `TrainingMediaInformationResourceItem`

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from TrainingMediaResource). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 trainingMediaResourceCategory

Inherited from TrainingMediaResource.

2.1.2.2 trainingMediaResourceDescription (zero, one or many)

Inherited from TrainingMediaResource.

2.1.2.3 trainingMediaResourceRationale (zero, one or many)

Inherited from TrainingMediaResource.

2.1.2.4 informationResourcePortion (zero, one or many)

informationResourcePortion is a description that provides further information on which part of an information item that is of interest for the development of instructional or learning assessment material.

2.2 TrainingMediaInformationResourceItem

2.2.1 Interface definition

TrainingMediaInformationResourceItem is a <<select>> interface that identifies items which can be selected as information assets during training development.

2.2.1.1 Class members

This <<select>> interface includes the following class members:

- DigitalFile. Refer to Chap 7.3.10.
- Document. Refer to Chap 7.3.11.
- DocumentIssue. Refer to Chap 7.3.11.
- Task. Refer to Chap 7.3.32.

2.3 TrainingMediaMaterialResource

2.3.1 Class definition

TrainingMediaMaterialResource is a TrainingMediaResource that exists in the form of physical item.

2.3.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of TrainingMediaMaterialResourceItem

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from TrainingMediaResource). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 trainingMediaResourceCategory
Inherited from TrainingMediaResource.

2.3.2.2 trainingMediaResourceDescription (zero, one or many)
Inherited from TrainingMediaResource.

2.3.2.3 trainingMediaResourceRationale (zero, one or many)
Inherited from TrainingMediaResource.

2.4 TrainingMediaMaterialResourceItem

2.4.1 Interface definition

TrainingMediaMaterialResourceItem is a <<select>> interface identifies physical items which can be considered during training development.

2.4.1.1 Class members
This <<select>> interface includes the following class members:

- BreakdownElement. Refer to Chap 7.3.3.
- PartAsDesigned. Refer to Chap 7.3.21.
- Product. Refer to Chap 7.3.23.
- ProductVariant. Refer to Chap 7.3.23.
- ResourceSpecification. Refer to Chap 7.3.27.

2.5 TrainingMediaResource

2.5.1 Class definition

TrainingMediaResource is a <<relationship>> that associates a selected LearningObjectiveTrainingMedia with an item which can be considered during training development for the associated LearningObjective. Refer to Chap 7.3.17.

2.5.1.1 Implementations
This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.5.2 Attributes

2.5.2.1 trainingMediaResourceCategory
trainingMediaResourceCategory is a classification that defines how the resource is to be regarded from the training material development perspective.

Valid values:

- `DEV` (SX001G:trainingDevelopmentResource)
- `IDEA` (SX001G:ideaOnTrainingProduct)
- `REQ` (SX001G:requiredTrainingProduct)

2.5.2.2 trainingMediaResourceDescription (zero, one or many)

`trainingMediaResourceDescription` is a description that provides further information about the identified resource in the context of the defined learning objective training media.

2.5.2.3 trainingMediaResourceRationale (zero, one or many)

`trainingMediaResourceRationale` is a description that gives more information on the justification for identifying the resource in the context of the defined learning objective training media

Chapter 7.3.43

Units of functionality - UoF Training Method

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.17	Units of functionality - UoF Learning Objective
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.41	Units of functionality - UoF Training Media and Fidelity
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

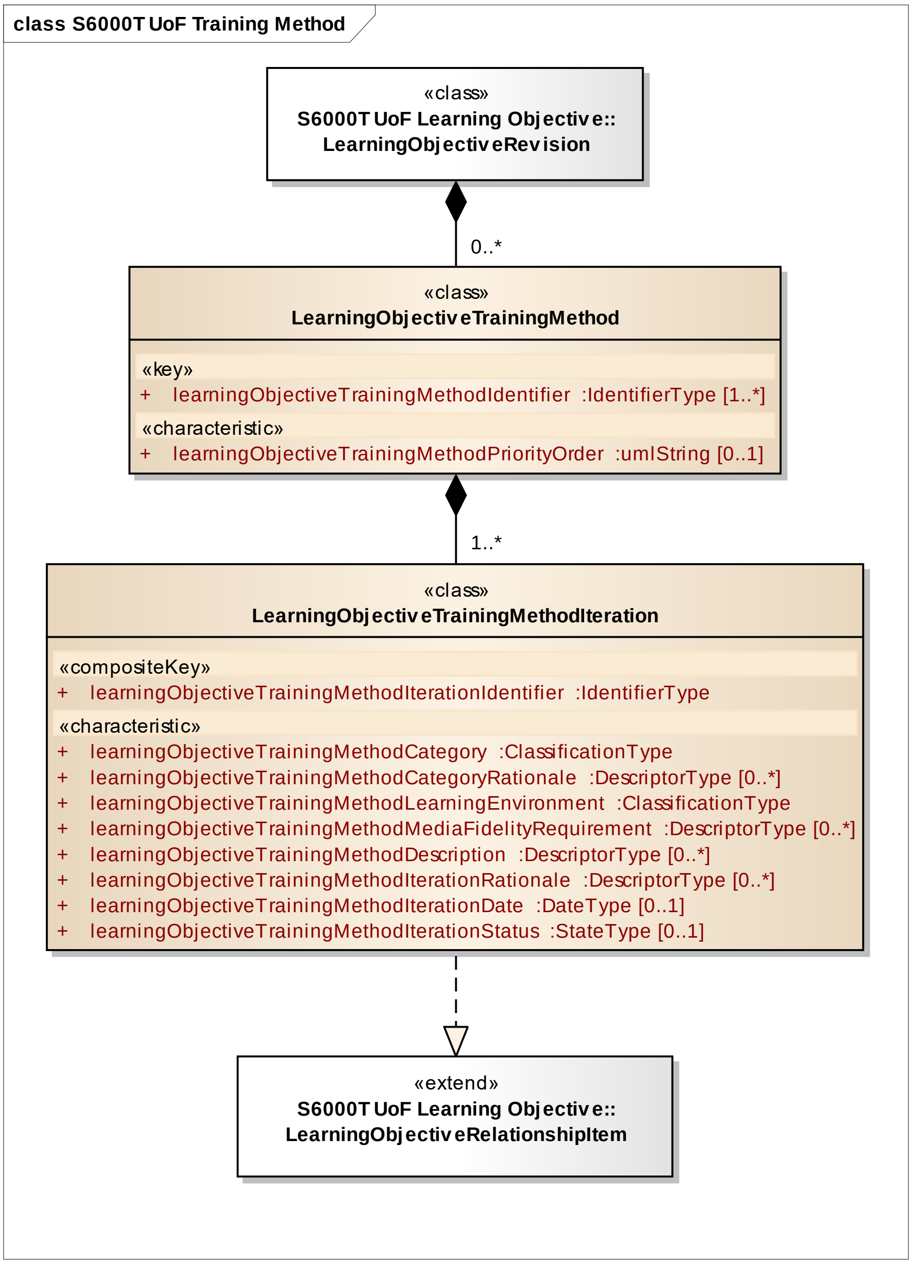
1 General

The Training Method UoF provides the capability to represent techniques used to support the acquisition of new knowledge, skills, and attitudes.

2 UoF Training Method

Key features of the UoF Training Method data model, (refer to Fig 1), are:

- training method is defined in the context of a learning objective
- a defined training method can in itself initiate additional learning objectives



ICN-S6000T-B6865-00100-001-01

Fig 1 UoF Training Method

2.1 LearningObjectiveTrainingMethod

2.1.1 Class definition

`LearningObjectiveTrainingMethod` is a `<<class>>` that identifies a technique which is suitable to support the acquisition of new knowledge, skills, and/or attitudes to meet the `LearningObjective`. Refer to Chap 7.3.17.

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of `LearningObjectiveRevision`

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- `ApplicabilityStatementItem`. Refer to Chap 7.3.2.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 learningObjectiveTrainingMethodIdentifier (one or many)

`learningObjectiveTrainingMethodIdentifier` is an identifier that establishes a unique designator for a `LearningObjectiveTrainingMethod` and to differentiate it from other instances of `LearningObjectiveTrainingMethod`.

Valid identifier class values:

- `ID` (SX001G:learningObjectiveTrainingMediaIdentifier)

2.1.2.2 learningObjectiveTrainingMethodPriorityOrder (zero or one)

`learningObjectiveTrainingMethodPriorityOrder` is a string value that ranks the `LearningObjectiveTrainingMethod` in relation to alternate instances of `LearningObjectiveTrainingMethod` defined for the same `LearningObjective`. Refer to Chap 7.3.17.

2.2 LearningObjectiveTrainingMethodIteration

2.2.1 Class definition

`LearningObjectiveTrainingMethodIteration` is a <<class>> representing an iteration applied to a `LearningObjectiveTrainingMethod`.

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of `LearningObjectiveTrainingMedia`. Refer to Chap 7.3.41.

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `LearningObjectiveRelationshipItem`. Refer to Chap 7.3.17.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 learningObjectiveTrainingMethodIterationIdentifier

`learningObjectiveTrainingMethodIterationIdentifier` is an identifier that establishes a unique designator for a `LearningObjectiveTrainingMethodIteration` and to differentiate it from other instances of `LearningObjectiveTrainingMethodIteration`.

Valid identifier class values:

- `ID` (SX001G:learningObjectiveTrainingMethodIterationIdentifier)

2.2.2.2 learningObjectiveTrainingMethodCategory

`learningObjectiveTrainingMethodCategory` is a classification that identifies the technique or format for training.

Valid values:

- `DEMO` (SX001G:demonstrationTrainingMethod)
- `INST` (SX001G:instructorFacilitatedTrainingMethod)
- `LECT` (SX001G:lectureTrainingMethod)
- `ROLE` (SX001G:rolePlayTrainingMethod)
- `SELF` (SX001G:selfDirectedTrainingMethod)
- `SOTJ` (SX001G:structuredOnTheJobTrainingMethod)
- `STU` (SX001G:studentCenteredTrainingMethod)

2.2.2.3 learningObjectiveTrainingMethodCategoryRationale (zero, one or many)

`learningObjectiveTrainingMethodCategoryRationale` is a description that gives more information on the justification for selecting the `learningObjectiveTrainingMethodCategory`.

2.2.2.4 learningObjectiveTrainingMethodLearningEnvironment

`learningObjectiveTrainingMethodLearningEnvironment` is a classification that defines the how training is to be delivered with respect to dependencies between trainees as well as between trainees and instructors.

Valid values:

- `A` SX001G:asyncronousLearningEnvironment
- `D` SX001G:distributedLearningEnvironment
- `S` SX001G:syncronousLearningEnvironment

2.2.2.5 learningObjectiveTrainingMethodMediaFidelityRequirement (zero, one or many)

`learningObjectiveTrainingMethodMediaFidelityRequirement` is a description that defines the real-world characteristics a training intervention must imitate or replicate in order for the training for the `LearningObjective` (refer to Chap 7.3.17) to be meaningful.

2.2.2.6 learningObjectiveTrainingMethodDescription (zero, one or many)

`learningObjectiveTrainingMethodDescription` is a description that provides further information on the identified training method.

2.2.2.7 learningObjectiveTrainingMethodIterationRationale (zero, one or many)
`learningObjectiveTrainingMethodIterationRationale` is a description that gives more information on the justification for revising the `LearningObjectiveTrainingMethod`.

2.2.2.8 learningObjectiveTrainingMethodIterationDate (zero or one)
`learningObjectiveTrainingMethodIterationDate` is a date that specifies when the `LearningObjectiveTrainingMethod` was revised.

2.2.2.9 learningObjectiveTrainingMethodIterationStatus (zero or one)
`learningObjectiveTrainingMethodIterationStatus` is a state that identifies the maturity of the `LearningObjectiveTrainingMethodIteration`.

Valid state values:

- `A` (SX001G:approvedStatus)
- `D` (SX001G:draftStatus)
- `IW` (SX001G:inWorkStatus)
- `C` (SX001G:cancelledStatus)
- `R` (SX001G:reviewedStatus)

Chapter 7.3.44

Units of functionality - UoF Training System Resource Requirement

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement

Chap No./Document No.	Title
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.6	Units of functionality - UoF Competency Definition
Chap 7.3.7	Units of functionality - UoF Course Definition
Chap 7.3.8	Units of functionality - UoF Course Element
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.21	Units of functionality - UoF Part Definition
Chap 7.3.23	Units of functionality - UoF Product
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.27	Units of functionality - UoF Resource Specification
Chap 7.3.32	Units of functionality - UoF Task
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

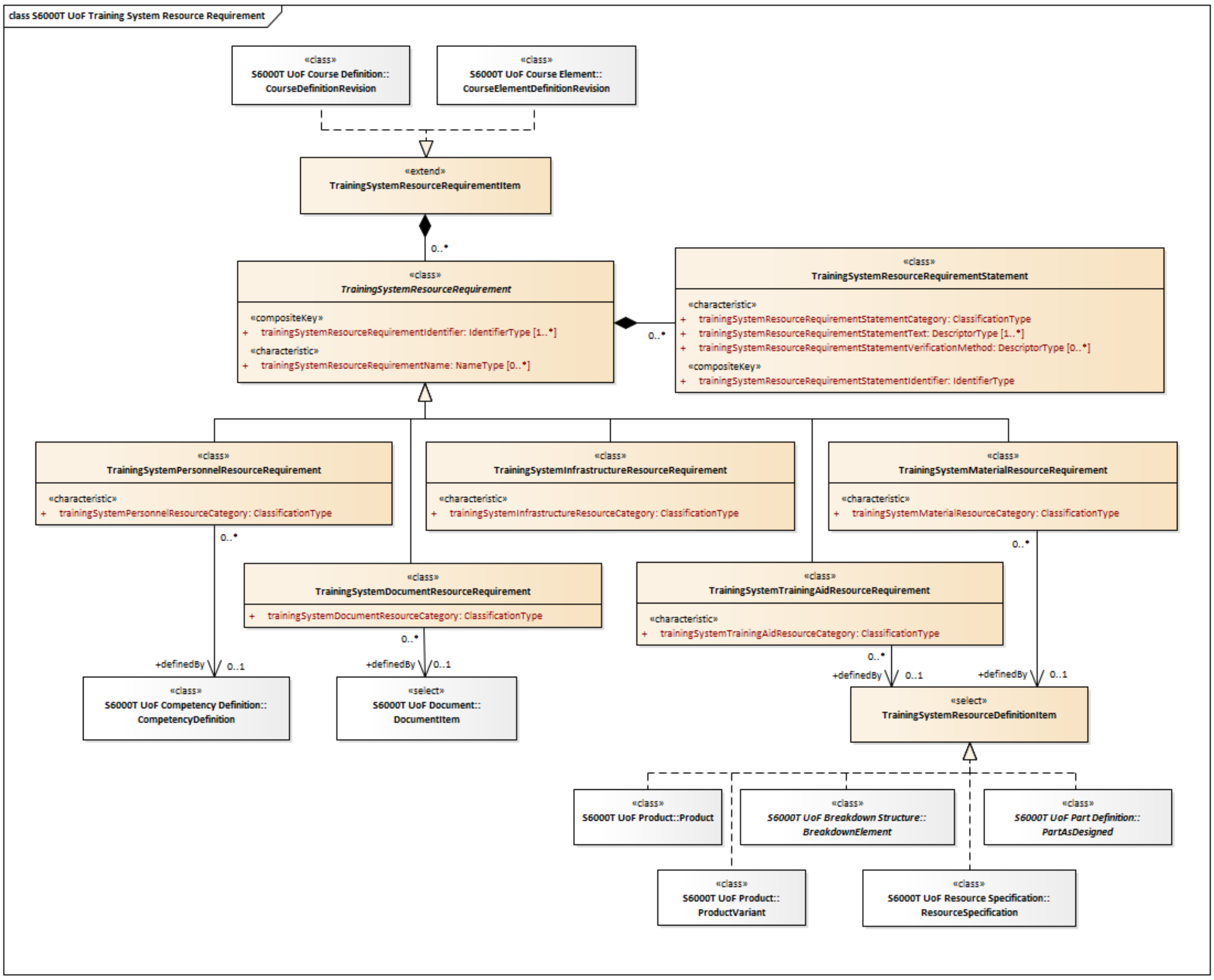
1 General

The Training System Resource Requirement UoF enables the documentation of means required to perform a specified amount of training.

2 UoF Training System Resource Requirement

Key features of the UoF Training System Resource Requirement data model, (refer to Fig 1), are:

- a training system resource requirement can either be specified for an overall course or per course element
- a training system resource requirement can be stated in terms of eg, budget, schedule, logistics and functional requirement
- a training system resource requirement must be defined as either a:
  - material resource requirement (eg, support equipment, spare parts, etc)
  - infrastructure resource (eg, hangar, power, computer, classroom, etc)
  - training aid resource (eg, simulator, part task trainer, etc)
  - personnel resource (eg, instructor, etc)
  - document resource (eg, technical manuals, etc)



ICN-S6000T-B6865-00104-002-01

Fig 1 UoF Training System Resource Requirement

2.1 TrainingSystemDocumentResourceRequirement

2.1.1 Class definition

`TrainingSystemDocumentResourceRequirement` is a `TrainingSystemResourceRequirement` that identifies a document which must be made available as training resource.

2.1.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of `TrainingSystemResourceRequirementStatement` (inherited from `TrainingSystemResourceRequirement`)
- A directed `definedBy` association with zero or one instance of `DocumentItem`. Refer to Chap 7.3.11.

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from TrainingSystemResourceRequirement). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 trainingSystemResourceRequirementIdentifier (one or many)

Inherited from TrainingSystemResourceRequirement.

2.1.2.2 trainingSystemResourceRequirementName (zero, one or many)

Inherited from TrainingSystemResourceRequirement.

2.1.2.1 trainingSystemDocumentResourceCategory

trainingSystemDocumentResourceCategory is a classification that identifies further specialization for a TrainingSystemDocumentResourceRequirement.

Valid values:

- TMD (SX001G: trainingManagementDocument)
- TSD (SX001G: trainingSupportDocument)

2.2 TrainingSystemInfrastructureResourceRequirement

2.2.1 Class definition

TrainingSystemInfrastructureResourceRequirement is a TrainingSystemResourceRequirement defined in terms of foundational systems and services.

2.2.1.1 Associations

This class has the following associations:

- An aggregate association with zero, one or many instances of TrainingSystemResourceRequirementStatement (inherited from TrainingSystemResourceRequirement)

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from TrainingSystemResourceRequirement). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2.2 Attributes

2.2.2.1 trainingSystemResourceRequirementIdentifier (one or many)
Inherited from TrainingSystemResourceRequirement.

2.2.2.2 trainingSystemResourceRequirementName (zero, one or many)
Inherited from TrainingSystemResourceRequirement.

2.2.2.3 trainingSystemInfrastructureResourceCategory
trainingSystemInfrastructureResourceCategory is a classification that identifies further specialization for a TrainingSystemInfrastructureResourceRequirement.

Valid values:

- CNWK (SX001G:communicationNetworkInfrastructureResource)
- COMP (SX001G:computerInfrastructureResource)
- DOCK (SX001G:dockInfrastructureResource)
- DRYD (SX001G:dryDockInfrastructureResource)
- GAR (SX001G:garageInfrastructureResource)
- HNG (SX001G:hangarInfrastructureResource)
- POW (SX001G:powerInfrastructureResource)
- TNWK (SX001G:transportNetworkInfrastructureResource)

2.3 TrainingSystemMaterialResourceRequirement

2.3.1 Class definition

TrainingSystemMaterialResourceRequirement is a TrainingSystemResourceRequirement that directly or indirectly refers to a physical item which is either defined in context of the Product (refer to Chap 7.3.23) structure or is defined as a resource in a Task (refer to Chap 7.3.32) to be performed in the context of the Product.

2.3.1.1 Associations
This class has the following associations:

- An aggregate association with zero, one or many instances of TrainingSystemResourceRequirementStatement (inherited from TrainingSystemResourceRequirement)
- A directed definedBy association with zero or one instance of TrainingSystemResourceDefinitionItem

2.3.1.2 Implementations
This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from TrainingSystemResourceRequirement). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.3.2 Attributes

2.3.2.1 trainingSystemResourceRequirementIdentifier (one or many)
Inherited from `TrainingSystemResourceRequirement`.

2.3.2.2 trainingSystemResourceRequirementName (zero, one or many)
Inherited from `TrainingSystemResourceRequirement`.

2.3.2.3 trainingSystemMaterialResourceCategory
`trainingSystemMaterialResourceCategory` is a classification that defines the role of the `TrainingSystemMaterialResourceRequirement` in the context of training.

Valid values:

- `C` (SX001G:consumablePart)
- `E` (SX001G:expendableSparePart)
- `EP` (SX001G:expendablePersonalProtectionPart)
- `HT` (SX001G:standardHandTool)
- `IT` (SX001G:informationTechnologyPart)
- `PA` (SX001G:packagingPart)
- `PE` (SX001G:personalProtectionPart)
- `PR` (SX001G:productProtectionPart)
- `R` (SX001G:repairableSparePart)
- `M` (SX001G:rawMaterial)
- `S` (SX001G:sparePart)
- `SE` (SX001G:supportEquipment)
- `SSE` (SX001G:safetyRelatedSupportEquipment)

2.4 TrainingSystemPersonnelResourceRequirement

2.4.1 Class definition

`TrainingSystemPersonnelResourceRequirement` is a `TrainingSystemResourceRequirement` defined in terms of manpower.

2.4.1.1 Associations
This class has the following associations:

- An aggregate association with zero, one or many instances of `TrainingSystemResourceRequirementStatement` (inherited from `TrainingSystemResourceRequirement`)
- A directed `definedBy` association with zero or one instance of `CompetencyDefinition`. Refer to Chap 7.3.6.

2.4.1.2 Implementations
This class implements the following `<<extend>>` interfaces:

- `ApplicabilityStatementItem` (inherited from `TrainingSystemResourceRequirement`). Refer to Chap 7.3.2.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 trainingSystemResourceRequirementIdentifier (one or many)
Inherited from `TrainingSystemResourceRequirement`.

2.4.2.2 trainingSystemResourceRequirementName (zero, one or many)
Inherited from `TrainingSystemResourceRequirement`.

2.4.2.1 trainingSystemPersonnelResourceCategory
`trainingSystemPersonnlResourceCategory` is a classification that identifies further specialization for a `TrainingSystemPersonnelResourceRequirement`.

Valid values:

- `SUP` (SX001G:supportPersonnelTrainingSystemResourceRequirement)
- `INS` (SX001G: instructorPersonnelTrainingSystemResourceRequirement)
- `MGT` (SX001G: managementPersonnelTrainingSystemResourceRequirement)

2.5 TrainingSystemResourceDefinitionItem

2.5.1 Interface definition

`TrainingSystemResourceDefinitionItem` is a `<<select>>` interface that identifies items which can be used as training aids or as training material resources.

2.5.1.1 Class members
This `<<select>>` interface includes the following class members:

- `BreakdownElement`. Refer to Chap 7.3.3.
- `PartAsDesigned`. Refer to Chap 7.3.21.
- `Product`. Refer to Chap 7.3.23.
- `ProductVariant`. Refer to Chap 7.3.23.
- `ResourceSpecification`. Refer to Chap 7.3.27.

2.6 TrainingSystemResourceRequirement

2.6.1 Class definition

`TrainingSystemResourceRequirement` is a `<<class>>` that identifies means that must be available to perform a specified amount of training.

2.6.1.1 Associations
This class has the following associations:

- An aggregate association with zero, one or many instances of `TrainingSystemResourceRequirementStatement`

2.6.1.2 Implementations
This class implements the following `<<extend>>` interfaces:

- `ApplicabilityStatementItem`. Refer to Chap 7.3.2.
- `DocumentReferencingItem` (inherited from `BaseObject`). Refer to Chap 7.3.11.
- `ProjectSpecificExtensionItem` (inherited from `BaseObject`). Refer to SX002D.
- `RemarkItem` (inherited from `BaseObject`). Refer to Chap 7.3.26.

2.6.2 Attributes

2.6.2.1 trainingSystemResourceRequirementIdentifier (one or many)

trainingSystemResourceRequirementIdentifier is an identifier that establishes a unique designator for a TrainingSystemResourceRequirement and to differentiate it from other instances of TrainingSystemResourceRequirement.

Valid identifier class values:

- ID (SX001G:trainingSystemResourceRequirementIdentifier)

2.6.2.2 trainingSystemResourceRequirementName (zero, one or many)

trainingSystemResourceRequirementName is a name by which the TrainingSystemResourceRequirement is known and can be easily referenced.

2.7 TrainingSystemResourceRequirementItem

2.7.1 Interface definition

TrainingSystemResourceRequirementItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

2.7.1.1 Associations

The TrainingSystemResourceRequirementItem <<extend>> interface has the following associations:

- An aggregate association with zero, one or many instances of TrainingSystemResourceRequirement

2.7.1.2 Class members

This <<extend>> interface includes the following class members:

- CourseDefinitionRevision. Refer to Chap 7.3.7.
- CourseElementDefinitionRevision. Refer to Chap 7.3.8.

2.8 TrainingSystemResourceRequirementStatement

2.8.1 Class definition

TrainingSystemResourceRequirementStatement is a <<class>> that expresses a single documented aspect which must be fulfilled by an item in order to qualify as the training system resource.

2.8.1.1 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.8.2 Attributes

2.8.2.1 trainingSystemResourceRequirementStatementIdentifier

trainingSystemResourceRequirementStatementIdentifier is an identifier that establishes a unique designator for a TrainingSystemResourceRequirementStatement and to differentiate it from other instances of TrainingSystemResourceRequirementStatement.

Valid identifier class values:

- ID (SX001G:trainingSystemResourceRequirementStatementIdentifier)

2.8.2.2 trainingSystemResourceRequirementStatementCategory
trainingSystemResourceRequirementStatementCategory is a classification that identifies further specialization for a TrainingSystemResourceRequirementStatement.

Valid values:

- B (SX001G:budgetTrainingSystemResourceRequirement)
- F (SX001G:functionalTrainingSystemResourceRequirement)
- L (SX001G:logisticsTrainingSystemResourceRequirement)
- S (SX001G:scheduleTrainingSystemResourceRequirement)

2.8.2.3 trainingSystemResourceRequirementStatementText (one or many)
trainingSystemResourceRequirementStatementText is a description that contains the text that specifies the detailed requirement.

2.8.2.4 trainingSystemResourceRequirementStatementVerificationMethod (zero, one or many)
trainingSystemResourceRequirementStatementVerificationMethod is a description that provides further information on the process to be used to evaluate whether a defined training system resource complies with the TrainingSystemResourceRequirementStatement.

2.9 TrainingSystemTrainingAidResourceRequirement

2.9.1 Class definition

TrainingSystemTrainingAidResourceRequirement is a TrainingSystemResourceRequirement that defines resources which are peculiar to training.

2.9.1.1 Associations
This class has the following associations:

- An aggregate association with zero, one or many instances of TrainingSystemResourceRequirementStatement (inherited from TrainingSystemResourceRequirement)
- A directed definedBy association with zero or one instance of TrainingSystemResourceDefinitionItem

2.9.1.2 Implementations
This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem (inherited from TrainingSystemResourceRequirement). Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.9.2 Attributes

2.9.2.1 trainingSystemResourceRequirementIdentifier (one or many)
Inherited from TrainingSystemResourceRequirement.

2.9.2.2 trainingSystemResourceRequirementName (zero, one or many)
Inherited from TrainingSystemResourceRequirement.

2.9.2.3 trainingSystemTrainingAidResourceCategory
trainingSystemTrainingAidResourceCategory is a classification that identifies further specialization for a TrainingSystemTrainingAidResourceRequirement.

Valid values:

- SIM (SX001G:simulator)
- PTT (SX001G:partTaskTrainer)

Chapter 7.3.45

Units of functionality - UoF Warning Caution Train Prioritization

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.14	Units of functionality - UoF Job Duty
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.32	Units of functionality - UoF Task
Chap 7.3.33	Units of functionality - UoF Task Knowledge Skill and Attitude
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

1 General

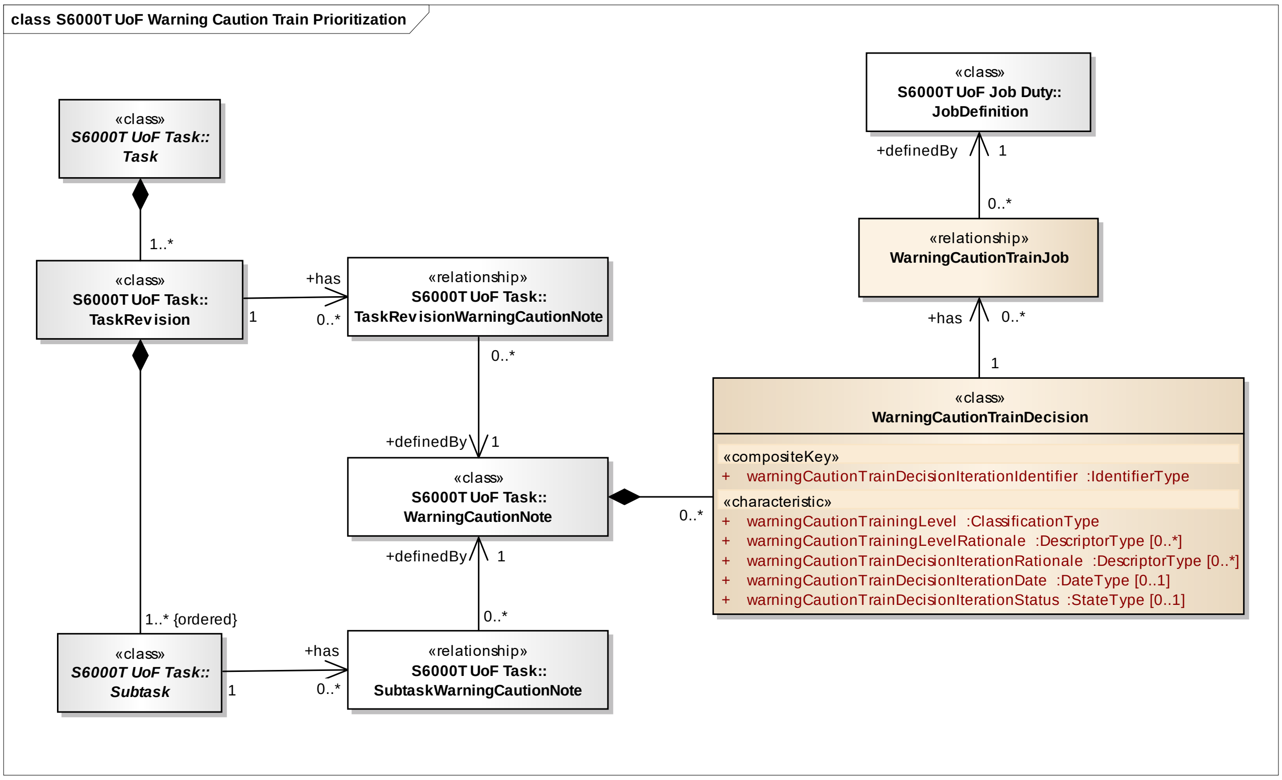
The Warning Caution Train Prioritization UoF provides the capability to define training prioritization for a warning or caution.

Note

The decision on task train prioritization can be iterated for a warning caution. For example, feedback on actual performance in the field can result in iterating the evaluation of the warning caution train prioritization.

2 UoF Warning Caution Train Prioritization

A key feature of the UoF Warning Caution Train Prioritization data model, (refer to Fig 1), is that the decision on train prioritization can be iterated for a warning caution. For example, feedback on actual performance in the field can result in iterating the evaluation of the warning caution train prioritization.



ICN-S6000T-B6865-00045-002-01

Fig 1 UoF Warning Caution Train Prioritization

2.1 WarningCautionTrainDecision

2.1.1 Class definition

`WarningCautionTrainDecision` is a `<<class>>` that defines how a `WarningCautionNote` (refer to Chap 7.3.32) is to be managed from a training perspective.

2.1.1.1 Associations

This class has the following associations:

– A directed `has` association with zero, one or many instances of `WarningCautionTrainJob`

2.1.1.2 Implementations

This class implements the following `<<extend>>` interfaces:

– `ApplicabilityStatementItem`. Refer to Chap 7.3.2.

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- KnowledgeSkillAttitudeRequirementItem. Refer to Chap 7.3.33.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.1.2 Attributes

2.1.2.1 warningCautionTrainDecisionIterationIdentifier

warningCautionTrainDecisionIterationIdentifier is an identifier that establishes a unique designator for a WarningCautionTrainDecision and to differentiate it from other instances of WarningCautionTrainDecision.

Valid identifier class values:

- ID (SX001G:warningCautionTrainDecisionIterationIdentifier)

2.1.2.2 warningCautionTrainingLevel

warningCautionTrainingLevel is a classification that identifies the priority for training for the performer to deal with a specific safety aspect.

Note

Training level is often referred to as training priority.

Valid values:

- 1 (SX001G:veryHighPriorityWarningCautionTraining)
- 2 (SX001G:highPriorityWarningCautionTraining)
- 3 (SX001G:moderatePriorityWarningCautionTraining)
- 4 (SX001G:lowPriorityWarningCautionTraining)
- 5 (SX001G:noFormalTrainingWarningCautionTraining)

2.1.2.3 warningCautionTrainingLevelRationale, zero, one or many

warningCautionTrainingLevelRationale is a description that gives more information on the justification for selecting the defined warningCautionTrainingLevel.

2.1.2.4 warningCautionTrainDecisionIterationRationale, zero, one or many

warningCautionTrainDecisionIterationRationale is a is a description that gives more information on the justification for iterating the decision on further training analysis for the warning caution.

2.1.2.5 warningCautionTrainDecisionIterationDate, zero or one

warningCautionTrainDecisionIterationDate is a date that specifies when the decision on further training analysis for the warning caution was revised.

2.1.2.6 warningCautionTrainDecisionIterationStatus, zero or one

warningCautionTrainDecisionIterationStatus is a state that identifies the maturity of the WarningCautionTrainDecision.

Valid state values:

- A (SX001G:approvedStatus)
- D (SX001G:draftStatus)

- IW (SX001G:inWorkStatus)
- C (SX001G:cancelledStatus)
- R (SX001G:reviewedStatus)

2.2 WarningCautionTrainJob

2.2.1 Class definition

WarningCautionTrainJob is a <<relationship>> that associates a WarningCautionTrainDecision with a JobDefinition (refer to Chap 7.3.14) which is affected by the train prioritization decision.

2.2.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of JobDefinition. Refer to Chap 7.3.14.

2.2.1.2 Implementations

This class implements the following <<extend>> interfaces:

- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

Chapter 7.3.46

Units of functionality - UoF Zone Element

Table of contents

Page

List of tables

List of figures

References

Table 1 References

Chap No./Document No.	Title
Chap 7.3.2	Units of functionality - UoF Applicability statement
Chap 7.3.3	Units of functionality - UoF Breakdown structure
Chap 7.3.4	Units of functionality - UoF Change Information
Chap 7.3.11	Units of functionality - UoF Document
Chap 7.3.26	Units of functionality - UoF Remark
Chap 7.3.28	Units of functionality - UoF Security Classification
SX001G	Glossary for the S-Series IPS specifications
SX002D	Common data model for the S-Series IPS specifications

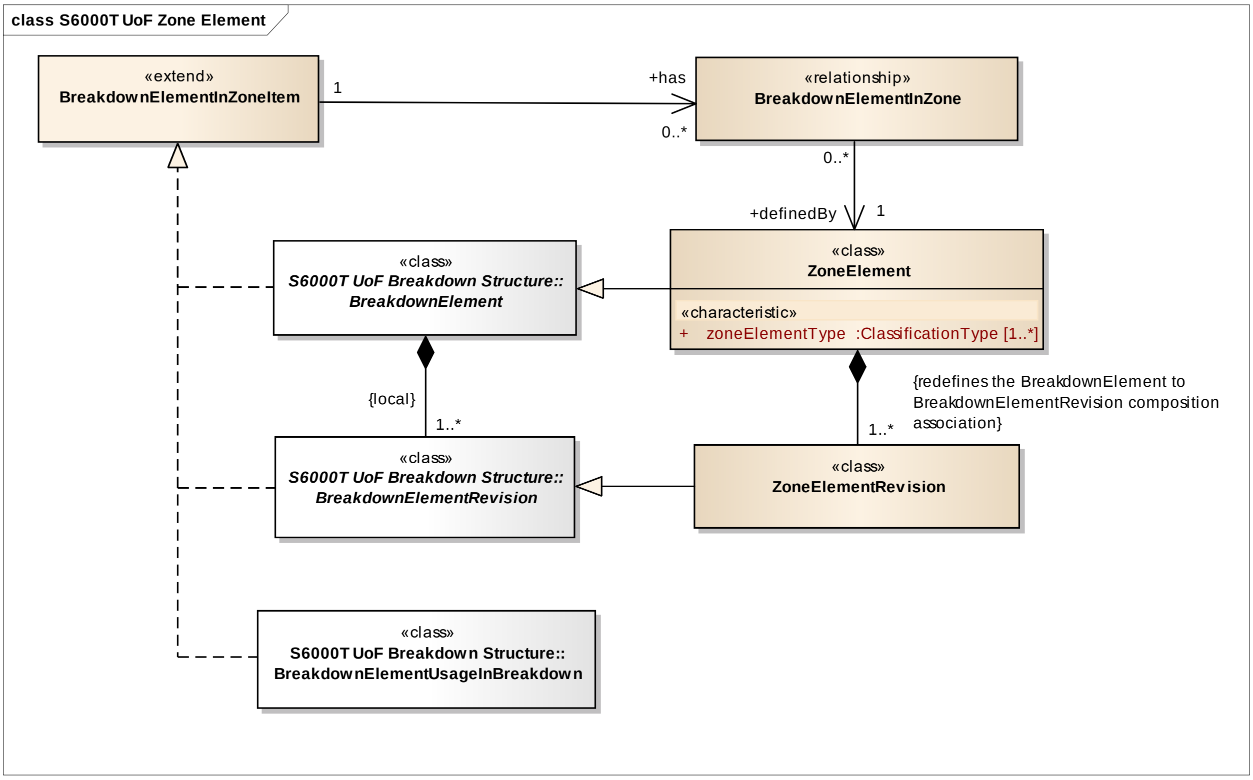
1 General

The Zone Element UoF defines the characteristics that are unique for a breakdown element that represents a three-dimensional space related to a Product.

2 UoF Zone Element

Key features of the UoF Zone Element data model, (refer to Fig 1), are:

- ZoneElement is a specialization of BreakdownElement, which means that wherever BreakdownElement is being used in the data model, ZoneElement can be used instead
- ZoneElementRevision is a specialization of BreakdownElementRevision, which means that wherever BreakdownElementRevision is being used in the data model, ZoneElementRevision can be used instead
- an instance of ZoneElement can have one or more associated BreakdownElements (including specific revisions and usages thereof) which are in the three-dimensional space that the ZoneElement represents



ICN-S6000T-B6865-00033-002-01

Fig 1 UoF Zone Element

2.1 BreakdownElementInZone

2.1.1 Class definition

BreakdownElementInZone is a <<relationship>> where a BreakdownElementInZoneItem relates to the ZoneElement where it is located.

Source:

- SX001G

2.1.1.1 Associations

This class has the following associations:

- A directed definedBy association with one instance of ZoneElement

2.1.1.2 Implementations

This class implements the following <<extend>> interfaces:

- ApplicabilityStatementItem. Refer to Chap 7.3.2.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.2 BreakdownElementInZoneItem

2.2.1 Interface definition

BreakdownElementInZoneItem is an <<extend>> interface that provides its associated data model to those classes that implement it.

Source:

- SX001G

2.2.1.1 Associations

The BreakdownElementInZoneItem <<extend>> interface has the following associations:

- A directed has association with zero, one or many instances of BreakdownElementInZone

2.2.1.2 Class members

Classes that implement the BreakdownElementInZoneItem <<extend>> interface are:

- BreakdownElement. Refer to Chap 7.3.3.
- BreakdownElementRevision. Refer to Chap 7.3.3.
- BreakdownElementUsageInBreakdown. Refer to Chap 7.3.3.

2.3 ZoneElement

2.3.1 Class definition

ZoneElement is a BreakdownElement (refer to Chap 7.3.3) that represents a three-dimensional space related to a Product.

Note

A zone can also represent a work area such as a mechanical workshop onboard a ship.

Source:

- SX001G

2.3.1.1 Associations

This class has the following associations:

- An aggregate association with one or many instances of ZoneElementRevision

2.3.1.2 Implementations

This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem (inherited from BreakdownElement)
  DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.
- SecurityClassificationItem (inherited from BreakdownElement). Refer to Chap 7.3.28.

2.3.2 Attributes

2.3.2.1 breakdownElementIdentifier (one or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.3.2.2 breakdownElementName (zero, one or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.3.2.3 breakdownElementEssentiality (zero or many)
Inherited from BreakdownElement. Refer to Chap 7.3.3.

2.3.2.4 zoneElementType (one or many)
zoneElementType is a classification that identifies further specialization for a ZoneElement.

Valid values:

- W (SX001G:productWorkArea)
- Z (SX001G:productZone)

Source:

- SX001G

2.4 ZoneElementRevision

ZoneElementRevision is a BreakdownElementRevision (refer to Chap 7.3.3) representing an iteration applied to a ZoneElement.

2.4.1.1 Implementations
This class implements the following <<extend>> interfaces:

- BreakdownElementInZoneItem (inherited from BreakdownElementRevision).
- ChangeControlledItem (inherited from BreakdownElementRevision). Refer to Chap 7.3.4.
- DocumentReferencingItem (inherited from BaseObject). Refer to Chap 7.3.11.
- ProjectSpecificExtensionItem (inherited from BaseObject). Refer to SX002D.
- RemarkItem (inherited from BaseObject). Refer to Chap 7.3.26.

2.4.2 Attributes

2.4.2.1 breakdownElementRevisionIdentifier
Inherited from BreakdownElementRevision. Refer to Chap 7.3.3.

2.4.2.2 breakdownElementDescription (zero, one or many)
Inherited from `BreakdownElementRevision`. Refer to Chap 7.3.3.

2.4.2.3 breakdownElementRevisionRationale (zero, one or many)
Inherited from `BreakdownElementRevision`. Refer to Chap 7.3.3.

2.4.2.4 breakdownElementRevisionDate (zero or one)
Inherited from `BreakdownElementRevision`. Refer to Chap 7.3.3.

2.4.2.5 breakdownElementRevisionStatus (zero or one)
Inherited from `BreakdownElementRevision`. Refer to Chap 7.3.3.

End of data module

Chapter 8

Terms, abbreviations, and acronyms

Table of contents

Chapter 8.1

Terms, abbreviations, and acronyms - Glossary of terms

Table of contents

List of tables

References

Table 1 References

Chap No./Document No.	Title
None	

1 Glossary of terms

Term	Definition
Action verb	Action verb conveys activity or behaviors and reflects the type of competency or performance that is to occur. Example: Analyze, place, cut, drive, open, and hold **Note** Action verbs reflect behaviors that are measurable, observable, verifiable, and reliable.
Active learning	Active learning defines an approach to learning in which students engage the material they study through reading, writing, talking, listening, and reflecting. **Note** Active learning focuses the responsibility of learning on the learners.
Affective taxonomy	Affective taxonomy is a classification that defines the task's level of sophistication with respect to emotional, motivational and/or social aspects.
Asynchronous learning	Asynchronous learning defines an approach to learning that does not occur in the same place or at the same time.

Term	Definition
	Note Asynchronous learning is usually done in the form of digital and online learning. Example: Prerecorded video lessons or game-based learning that trainees complete on their own that is not being delivered in person or in real time.
Attitude	Attitude represents emotions required to perform a task or sub task. Example: Feelings, values, appreciation, enthusiasm, motivations
Behavior condition	Behavior condition is a description that expresses the environmental, physical, and psychological circumstances under which a behavior is portrayed.
Blended learning	Blended learning defines an approach that combines face-to-face classroom approaches with technology-delivered instruction that can be delivered either in a resident or non-resident environment to form an integrated instructional approach.
Bloom's taxonomy	Bloom’s taxonomy defines three hierarchical models used to classify ng objectives into levels of complexity and specificity. The three lists cover the learning objectives in cognitive, affective, and psychomotor domains.
Cognitive taxonomy	Cognitive taxonomy is a classification that defines the task's level of sophistication with respect to mental, cognitive, and/or logical aspects. **Note** Cognitive taxonomy includes the recall or recognition of specific facts, procedural patterns, and concepts that serve in the development of knowledge and mental skills.
Collective task	Collective task is a task that requires more than one person to complete, with each person performing a discrete part of the task or requires simultaneous performance of task steps in different locations or contains so many skills that one person cannot perform it in a timely or effective manner.
Competency	Competency represents measurable or observable Knowledge, Skills, and Attitudes (KSA) held by a person.
Competency definition	Competency definition is a term that defines measurable or observable KSA necessary for successful performance by a person in a given context

Term	Definition
Competency framework	Competency Framework is the authority that defines a competency definition.
Content analysis	Content analysis is a procedure for organizing narrative and qualitative data into emerging themes and concepts. **Note** Content analysis is usually associated with a quantitative form of analysis in which the themes are counted or measured.
Course	Course is a complete series of instructional units including lessons and other instructional segments.
Course description	Curriculum description is a description of a training curriculum and its contained available training courses and their respective requirements.
Course evaluation result	Course evaluations result represent a trainee survey used to measure training course performance.
Course phase	Course phase is a major part of a course which can be taught at different times due to resource and/or other constraints. **Note** Course phases are required as a necessary break-up of a course due to time, location, equipment, facility constraints, or training media.
Criterion	Criterion represents a principle or standard by which something can be judged or decided. In learning, the learning objective standard is the measure of trainee performance. In assessment validation, it is the standard against which assessment instruments are correlated to indicate the accuracy with which they predict human performance in some specific area. In evaluation, it is the measure used to determine the adequacy of a product, process, or behavior.
Curriculum	Curriculum defines a course (or set of courses) and their content offered by a formal training application.
DIF model	DIF model is a task selection model that identifies the Difficulty, Importance and Frequencies of a task.
Distributed Learning (DL)	Distributed learning represents the delivery of standardized individual, collective, and self-development training at the right place and right time, using multiple means and technologies, with synchronous and blended trainee-instructor interaction.
Educational taxonomy	Educational taxonomy is a classification system used to define and distinguish different levels of human cognition.

Term	Definition
	Example: Thinking, learning, and understanding. Reference: Bloom's taxonomy
Enabling Learning Objective (ELO)	ELO defines a subset of the KSA that trainees must reach to successfully complete the Terminal Learning Objective (TLO). **Note** ELOs allow the TLO to be broken down into smaller, more manageable objectives. An ELO supports the TLO and measures an element of the TLO, and addresses KSA gaps. ELOs are optional based on analysis of the TLO and when used, there must be a minimum of two.
Gap analysis	Gap analysis is an analysis activity that identifies the difference between current performance and the desired performance.
Human performance analysis	Human performance analysis is an analysis that looks at a user’s current performance and identifies whether the person is performing as desired. **Note** Human performance analysis is also known as KSA gap analysis.
Human performance deficiency	Human performance deficiency defines difference with a negative connotation, implying that the person is not meeting a known standard for performance.
Human performance measure	Human performance measure objectively observes and measures actions to determine if a task performer has performed the task to the prescribed standard. **Note** These measures are derived from the task performance steps during task analysis.
Individual task	An individual task is a task that requires one person to complete, with that person performing the entire task.
Instructor	Instructor is a person whose job it is to pass along to a trainee the required KSA required to successfully complete an activity.
Instructional designer	Instructional designer is a training role that designs and develops training content and other solutions to support the acquisition of new KSA.
Instructional material	Instructional material is material used by instructors/facilitators and/or learners in formal courses.

Term	Definition
	Example: Training aids, manuals, publications, and visual aids.
Instructional staff	Instructional staff is personnel that support and maintain a training system.
Job	Job represents a collection of unique, specific, related activities that fulfill a specific role or function critical to achievement of an organizations mission or operational objective.
Job aid	Job aid is a supporting item that facilitates the performance of a job.
Job analysis	Job analysis is a systematic study to decompose, structure, and comprehensively describe a job for the purpose of establishing training requirements.
Knowledge	Knowledge represents facts, information, and skills acquired by a person through experience or training.
Learning	Learning is the acquisition of knowledge or skills through experience, study, or formal instruction.
Learning gap	The gaps in KSA's of the target audience compared to the KSA's required for the performance of a task.
Learning objective	Learning objective is a statement that defines the expected goal of a learning activity in terms of demonstrable KSA. **Note** It is expressed in terms of condition, behavior and standard.
Lesson	Lesson represents an event where learning is intended to occur. **Note** During a lesson, trainees are taught about a particular subject or taught how to perform a particular activity. A lesson provides the instructional content for a lesson plan.
Lesson plan	Lesson plan represents detailed information used by instructors/facilitators to execute the instruction prescribed in one lesson within the prescribed time limits using the specified resources. **Note** A lesson plan includes the content and supporting information for only one lesson which supports the learning and assessment of one TLO.
Mastery	Mastery identifies possession of superior skills in the performance of an activity.

Term	Definition
Method of Instruction (MOI)	Method of Instruction is an activity used to facilitate the accomplishment of the learning objective(s). **Note** Specific methods of instruction require varying degrees of trainee participation. Selection of the best MOI requires consideration of the trainee, the content, the goals, the learning environment, the instructor/facilitator, and the available resources.
Mission analysis	Mission analysis is an analysis that identifies a Product operational scenario where a person, or body of persons, can be tasked to perform a service or carry out an activity.
Nonresident learning	Nonresident learning defines an approach to learning that is distributed to trainees for completion without the physical presence of an on-site instructor/facilitator, small group leader or otherwise designated trainer.
Occupational survey	Occupational survey is any job-related information regarding physical demands, environment conditions, education, training, and experience, as well as cognitive and mental requirements of a job.
Ohio State Systematic Curriculum and Instructional Development (SCID) DIF Task Model	SCID DIF Task Model is a method used to identify training prioritization for a task based on its identified difficulty, importance, and frequency classifications.
Optimum Class Size (OCS)	Optimum Class Size defines the appropriate number of trainees in a class that can be taught indefinitely with no degradation in the effectiveness of instruction.
Performance objective	Performance objective defines performance expected in the execution of a task (subtask or subtask step) expressed in terms of behavior conditions, behavior performance and behavior standard.
Personnel qualification standard	Personnel qualification standard is a qualification system of basic performance designed to establish the minimum level of competency required for a person to successfully perform in their job and proficiency requirement.
Performance standard	Performance standard description that expresses the expected quality and/or time constraints that must be met in the performance of the task or a learning objective. Performance standard defines the accepted proficiency level required to accomplish a task or learning objective. **Note** Performance standard is normally expressed in terms of quality or time constraints.

Term	Definition
	Note Performance standard is a statement that establishes criteria for how well a task or learning objective is performed.
Plan of Instruction (POI)	Plan of Instruction is a course control document used for every block of instruction within a training course. **Note** Plan of Instruction includes objectives of the instruction block, duration, support material, and guidance scope. **Note** Plan of Instruction is also called syllabus.
Product	Product represents a family of items which share the same underlying design purpose.
Product function	Product function represents activities performed by a Product to satisfy the requirements of a mission or operation.
Project	Project represents the overall set of IPS activities defined for a Product.
Project team	Project team is a group of persons who work together toward the successful completion of a stated project goal.
Psychomotor taxonomy	Psychomotor taxonomy is a classification that defines a level of sophistication with respect to physical, mechanical, and/or movement-based aspects.
Resident learning	Resident learning is an approach to learning that is presented, managed, and controlled by an on-site instructor or facilitator, small group leader, or otherwise designated trainer.
Risk management	Risk management is the process of identifying, assessing, and controlling risks arising from operational factors and making decisions that balance risk cost with benefits.
Scenario	Scenario is a graphic and narrative description of an operational environment, means, and events of a current or hypothetical situation that sets a context for specific activities and establishes the conditions to achieve desired outcomes.
Scope	Scope defines the area covered by an activity or topic written as a clear, concise statement to achieve an end state.
Self-paced learning	Self-paced learning defines an approach to learning where a person moves through the course at varying rates according to parameters established during validation.

Term	Definition
	Note Trainees move through the course individually at their own speed and are not dependent on group times.
Seminar	Seminar is a course of study for discussion and research under the guidance of a recognized expert.
Sharable Content Object (SCO)	SCO is a learning object representing the smallest unit of instruction that is launched and received from a Learning Management System (LMS). **Note** A SCO is any entity (digital or non-digital) which can be used, reused, or referenced during technology-supported learning.
Sharable Content Object Reference Model (SCORM)	SCORM specifies a framework for content that meets the following requirements for Web-based content: interoperability, accessibility, reusability, durability, maintainability, and adaptability.
Simulator	Simulator is a machine with a similar set of controls designed to provide a realistic imitation of the operation of a vehicle, aircraft, or other complex system, used for training purposes.
Skill	Skill represents the cognitive, psychomotor, and affective abilities in the execution or performance of an activity.
Skill level	Skill level represents the identified proficiency or ability typically required for successful performance at the grade with which the skill level is associated.
Small Group Instruction (SGI)	SGI is an instruction that uses small group processes, methods, and techniques to stimulate learning.
Subject Matter Expert (SME)	SME is a person with extensive experience in performing the job or task being analyzed, as well experience mastering the performance requirements of a job/task being trained. **Note** SME knowledge qualifies the person to assist in the development process (eg, consultation, review, analysis). Normally, SMEs instruct in their area of expertise.
Synchronous learning	Synchronous learning is training where education, instruction, and learning occur at the same time, but not in the same place. **Note** The term is most applied to various forms of televisual, digital, and online learning in which trainees learn from instructors, colleagues, or peers in real time, but not in person.

Term	Definition
	Example: Educational video conferences, interactive webinars, chat-based online discussions, and lectures that are broadcast at the same time they delivered would all be considered forms of synchronous learning.
Task	Task represents the specification of work to be done or undertaken.
Task analysis	Task analysis is the process to define and describe the best method and sequence of steps to complete a specific task.
Task difficulty	Task difficulty is a classification that identifies the complexity of a Task from a human performance perspective.
Task force	Task force is a group of people formed to carry out a specific project, or to solve a problem.
Task frequency	Task frequency specifies the rate of occurrence of a task.
Task importance	Task importance is a classification that identifies possible adverse effects that the performance of a task can have with respect to cost (damage) and availability for the item under analysis if not completed successfully.
Task inventory	Task inventory identifies a set of tasks for a given purpose.
Terminal learning objective (TLO)	TLO is the highest-level Learning Objective appropriate to the human performance requirements a trainee will accomplish when successfully completing the instruction. **Note** TLO is the performance required of the trainee to demonstrate appropriate KSA normally required at the lesson level.
Total Task Inventory (TTI)	Total task inventory is a comprehensive enumeration of all tasks an incumbent performs as part of a job.
Train target audience	Train target audience defines the person or group of persons involved in a training need's assessment or training program. **Note** Persons for whom the instructional or training materials are designed. Samples from this audience are used in evaluating training materials during their development. Also called target audience.
Train task selection model	Train task selection model is a method used to perform training task selection analysis.

Term	Definition
Trainer	Trainer is a term used in a corporate setting for an instructor.
Trainee	Trainee is a person undergoing training for a particular job or profession.
Trainee behavior	Trainee behavior specifies what a trainee must do to satisfy a performance requirement. **Note** Behavior can involve recall, manipulation, discrimination, problem-solving, performing a step-by-step procedure, or producing a product.
Trainee handout	Trainee handout represents a resource used to support training which can free trainees from excessive note taking and or provides supplemental information. Example: Booklet, schematic, circuit diagram, table, or similar material that augments the study guide, workbook, trainee text, or otherwise supports course objectives.
Training analysis	Training analysis is the first phase in the training Analysis, Design, Develop, Implement and Evaluate (ADDIE) process where training needs and requirements are identified.
Training analyst	Training analyst is a role that performs a systematic study to determine the requirements to ensure that the training needs of a Product are met.
Training assessment	Training assessment or examination is a series of questions or exercises to measure the KSA of a trainee or group of trainees. **Note** Training assessment determines if a trainee or group of trainees can accomplish the objective to the established standard.
Training design	Training design is a phase in the training ADDIE process where managers and training developers translate analysis data into an outline for learning, create a blueprint for learning product development, and determine the sequence and how to train.
Training development	Training development is a phase in the training ADDIE process that is the act of taking design outputs and expanding on the learning activities, refining the course management plan, refining the resources, and creating the learning products. **Note** Training development is the production phase of ADDIE.

Term	Definition
Training equipment	Training equipment is equipment uniquely developed and used to the conduct training or support a training system.
Training evaluation	Training evaluation is a phase in the training ADDIE process that is the quality control mechanism for learning and learning product development. **Note** Training evaluation is a systematic and continuous method to appraise the quality, effectiveness, and efficiency of a program, process, product, or procedure.
Training facility	Training facility is a facility used for the conduct of training programs or training activities.
Training feasibility analysis	Training feasibility analysis is a cost-benefit analysis that estimates of the cost of the training weighed against the possible benefits that could be achieved if training were conducted.
Training gap	The gap between the behavior, conditions and standards, required to perform a new or changed task, and any pre-existing LO included in a training curriculum.
Training implementation	Training implementation is a phase in the training ADDIE process that is the conduct and delivery of the course/event in accordance with how the course/event was designed.
Training level	Training level is a classification that identifies the priority of training for the performer to be able to carry out the task within its defined performance standard. **Note** Training level is often rated from one to five where one represents over training required and five represents formal training is not required.
Training media	Training media represents any means or instrument of communication that is used as a part of an instructional sequence to demonstrate or clarify course content and to facilitate learning or increase comprehension of course material. Example: Auditory media, visual media, and printed media
Training module	Training module is a group of multiple related lessons that promotes efficiency in the course structure.
Training need interview	Training need interview is the activity of asking questions to experts or performers to identify training needs.

Term	Definition
Training needs assessment	Training needs assessment is the activity to gather information about work requirements that can be resolved by training or other human interventions. **Note** Training needs assessment include performance analysis, target audience analysis, sorting training needs and wants, job analysis, and task analysis.
Training Needs Analysis (TNA)	TNA is the method of determining if a training need exists and, if it does, what training is required to complete the performance requirements of a Product.
Training audience description	Training audience description is a description that gives more information on the trainee audience for who the training program being developed. **Note** Training audience description can include training background, job experiences, etc.
Training Situation Analysis (TSA)	TSA is an analysis activity that is performed to define the need for training and to identify and evaluate possible alternative solutions.
Training Situation Analysis Report (TSAR)	TSAR is a formal report used to document the results of the training situation analysis.
Training situation document	Training situation document is a formal deliverable that details the structure/format of a TSAR.
Training solution	Any content accessible by a trainee at the time and place of need that includes a defined behavior(s), condition(s), standard(s), and method of measurement aligned to the KSA required to perform a task.
Training strategy	Training strategy is a blueprint for training development to meet the training needs of the organization and support mission accomplishment.
Training system organization	Training system organization is an organizational structure for the training organization.
Training task analysis	Training task analysis is a systematic study to determine the performance requirements of tasks, and subtasks, and to determine the KSA required in the performance of tasks and subtasks.
Training task list	Training task list is a complete enumeration of all tasks that require training.
Training technology assessment report	Training technology assessment report is part of the training situation analysis report that details the technology used to support a training system.
Training assessment validation	Training assessment validation is a process used to determine if a training assessment successfully measures the intended objectives.

Term	Definition
Training validation	Training validation is the process used to determine if new/revised courses and training products/materials accomplish their intended purpose efficiently and effectively. **Note** Training validation is the process used to determine if training accomplishes its intended purpose.

Chapter 8.2

Terms, abbreviations, and acronyms - Abbreviations and acronyms

Table of contents

Page

List of tables

References

Table 1 References

Chap No./Document No.	Title
None	

1 Abbreviations and acronyms

1.1 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.

1.2 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.

1.3 Tense and number

The same abbreviation must be used for all tenses, possessive cases, singular and plural forms of a given word.

1.4 Abbreviation and acronym list

Abbreviation/acronym	Definition
ADDIE	Analysis, Design, Development, Implementation, and Evaluation
A/V	Audio Visual
CAI	Computer Assisted Instruction
CBT	Computer Based Training
CONOPS	Concept of Operations
CRI	Criterion Referenced Instruction
DIF	Difficulty, Importance and Frequencies.
DL	Distributed Learning
ELO	Enabling Learning Objective
HFE	Human Factors Engineering
IETM	Interactive Electronic Technical Manual
IETP	Interactive Electronic Technical Publication
IG	Instructor Guide
ILT	Instructor Led Training
IMI	Interactive Multimedia Instruction
IPS	Integrated Product Support
ISD	Instructional System Design
ISHPO	In-Service Human Performance Optimization
ISP	Integrated Support Plan
ITP	Individual Training Plan
JA	Job Aid
JITT	Just In Time Training
JTA	Job/Task Analysis
KPP	Key Performance Parameter
KSA	Knowledge, Skills and Attitude
LO	Learning Objectives
LORA	Level of Repair Analysis
LP	Lesson Plan
LS	Learning Step/Activity
LSA	Logistical Support Analysis
MOI	Method of Instruction

Abbreviation/acronym	Definition
MTA	Maintenance Task Analysis
NA	Needs Assessment
OJT	On the Job Training
PA	Performance Assessment
PE	Practical Exercise
PO	Performance Objective
POI	Plan of Instruction
PSRS	Product Support Requirements Specification
QA	Quality Assurance
QC	Quality Control
SAT	Systems Approach to Training
SME	Subject Matter Expert
SSDD	Support System Design Description
TD	Training Development
TLO	Terminal Learning Objective
TM	Technical Manual
TNA	Training Needs Assessment (also Training Needs Analysis)
TNG	Training
TQR	Training Quality Report
TSA	Training Situation Analysis
TSAR	Training Situation Analysis Report
TSP	Training Support Package
WBT	Web Based Training