OCORA

Open CCS On-board Reference Architecture

Requirement Management

Guideline

Document ID: OCORA-TWS05-010

Version: 2.01

Release: R1

Date: 26.11.2021

Revision History

Version	Change Description	Initials	Date of change
1.0	Official version for OCORA Delta Release	RM	30.06.2021
2.01	Official version for OCORA Release R1	RM, TM	26.11.2021

Table of Contents

References

Reader's note: please be aware that the document ids in square brackets, e.g. [OCORA-BWS01-010], as per the list of referenced documents below, are used throughout this document to indicate the references to external documents. Wherever a reference to a TSI-CCS SUBSET is used, the SUBSET is referenced directly (e.g. SUBSET-026). OCORA always reference to the latest available official version of the SUBSET, unless indicated differently.

[OCORA-BWS01-010] – Release Notes

[OCORA-BWS01-020] – Glossary

[OCORA-BWS01-030] – Question and Answers

[OCORA-BWS01-040] – Feedback Form

[OCORA-BWS03-010] - Introduction to OCORA

[OCORA-BWS04-010] - Problem Statements

[OCORA-BWS08-010] - Methodology

[OCORA-BWS08-020] - Tooling

[OCORA-TWS04-011] – Functional Vehicle Adapter – Requirements

[OCORA-TWS05-020] – Stakeholder Requirements

[OCORA-TWS05-021] – Program Requirements

[OCORA-TWS05-022] – Design Requirements

[OCORA-TWS09-010] - Testing - Strategy

[EUROSPEC RM 2 0816] - EuroSpec Requirements Management, 2nd edition, August 2016

[EARS] - The Easy Approach to Requirements Syntax: The Definitive Guide

[ISO29148:2018] - Systems and software engineering

[ISO15288:2015] Systems and software engineering — System life cycle processes, 2015

1 Introduction

1.1 Purpose of the document

The purpose of this document is to to define OCORAs requirements management.
This document is addressed to experts in the CCS domain and to any other person, interested in the OCORA concepts for on-board CCS. The reader is invited to provide feedback to the OCORA collaboration and can, therefore, engage in shaping OCORA. Feedback to this document and to any other OCORA documentation can be given by using the feedback form [OCORA-BWS01-040] .

If you are a railway undertaking, you may find useful information to compile tenders for OCORA compliant CCS building blocks, for tendering complete CCS system, or also for CCS replacements for functional upgrades or for life-cycle reasons.

If you are an organisation interested in developing CCS building blocks according to the OCORA standard, information provided in this document can be used as input for your development.

1.2 Applicability of the document

The document is currently considered informative but may become a standard at a later stage for OCORA compliant on-board CCS solutions. Subsequent releases of this document will be developed based on a modular and iterative approach, evolving within the progress of the OCORA collaboration.

1.3 Context of the document

This document is published as part of the OCORA Release R1, together with the documents listed in the release notes [OCORA-BWS01-010] . Before reading this document, it is recommended to read the Release Notes [OCORA-BWS01-010]. If you are interested in the context and the motivation that drives OCORA we recommend to read the Introduction to OCORA [OCORA-BWS03-010], and the Problem Statements [OCORA-BWS04-010]. The reader should also be aware of the Glossary [OCORA-BWS01-020] and the Question and Answers [OCORA-BWS01-030].

2 OCORAs Requirements Management

2.1 Introduction

Requirements are an important deliverable of OCORA. Therefore a structured requirements management approach is mandatory. Since we're facing a ‘brown-field’ approach in a regulated environment, many requirements are already given and different solutions for requirement management are already applied. In OCORA we are aiming for a robust requirement management process, which may add complexity at the beginning but enables innovation and transparency by applying the following characteristics.

- Full traceability of external (i.e. from outside OCORA) and internal requirements to anticipate their changes but also to trace back inherited problems during implementation
- Strict focus on business objectives for requirements and design decisions
- Clustering to building blocks to support modularity and the discussion with potential suppliers

OCORAs requirements management guideline is based on EuroSpec’s Requirement Management, [EUROSPEC RM 2 0816]

2.2 Expected Results

OCORA aims to provide tender templates for the next generation CCS On-board solutions. Over time these requirements will be enriched with return on experience from past procurements.

OCORA requirements will be provided in a structured format. This allows procurement projects to pick the suitable set of requirements and OCORA to manage the generic requirements catalogue over time, including traced activities, such as:

- adding (e.g. new applications and functionality),
- changing (e.g. update of norms and standards) and
- releasing (e.g. superseded functionality)

According to the OCORA Tooling definition, ref. [OCORA-BWS08-020] all OCORA Requirements will be managed in Polarion.

Modelling artefacts exported from Capella will be reviewed and traced in Polarion even though they origin from Capella and any changes need to be performed in Capella (Master).

3 Definitions

3.1 Requirements Structure

For a program like OCORA, the transparent management of requirements is essential. The structure of the requirements can be found in Figure 1

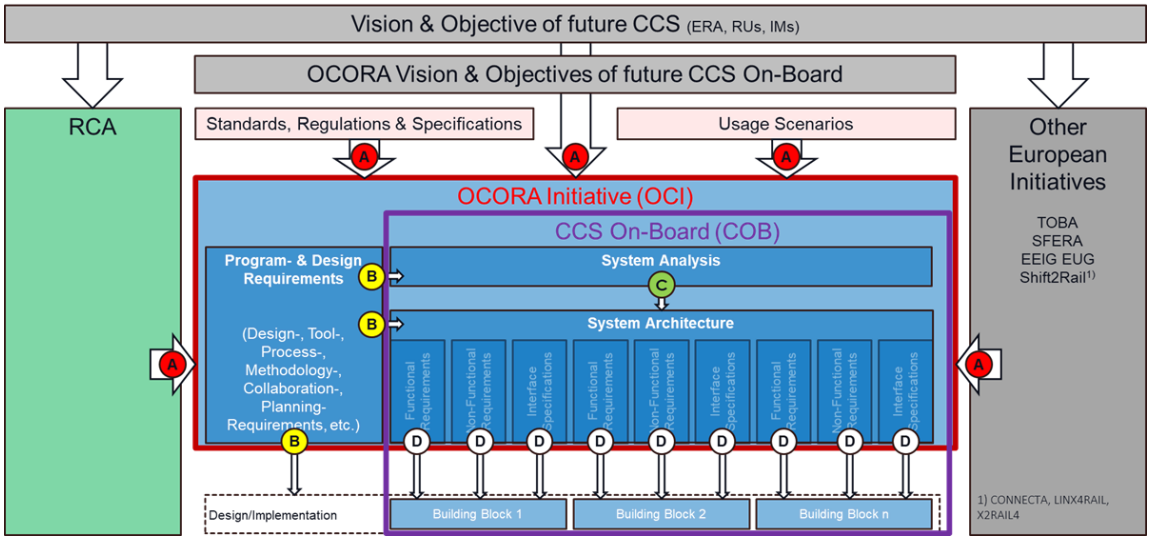

Figure *1* OCORA Requirements Structure

In a nutshell, the **Stakeholder Requirements (A)** are the foundation of OCORA requirements. They contain all requirements towards the OCORA Program and the envisages CCS On-board System. In essence they include:

- OCORA Vision and Objective of future CCS On-board as described in [OCORA-BWS03-010]
- Regulations / Norms and Standards
- Operational Scenarios
- Requirements from external Stakeholders (e.g. RCA, ER JU / Shift2Rail, EULYNX, EUG, TOBA, SFERA, etc.)

Program- & Design Requirements (B) are focusing on how the OCORA program defines tools, processes, methodology, collaboration, planning and applicable design rules. They are to be used within the program and to be considered during the system analysis and the system design/architecture work.

System Requirements (C) are defining the CCS On-board system, hence they describe how the system is developed in the MBSE System Analysis (RCA & OCORA), considering the A- and B-Level Requirements.

The **Building Block Requirements (D)** regarding the OCORA building blocks, are developed in the MBSE System Architecture (logical / physical), considering the MBSE System Analysis. The resulting documentation form the OCORA inputs for tender templates, together with the applicable program requirements.

3.2 Requirements Characteristics

3.2.1 Quality criteria for individual requirements

Following ISO 29148, ref. [ISO29148:2018] the characteristics of individual requirements are defined as:

Characteristics	Definition
Necessary	• The requirement defines an essential capability, characteristic, constraint and or quality factor • If it is left out, a deficiency in the product will exist
Unambiguous	• The requirement shall not be open for interpretation
Consistent	• The requirement is free of conflicts with other requirements in the specification • The requirement is free of conflicts in itself
Complete	• All necessary information is provided in the requirement
Singular	• The requirement statement includes only one requirement
Verifiable	• The requirement has the means to prove that the system satisfies the specified requirement
Traceable	• The requirement has a relation to its origin (e.g. a requirement or need from a higher level), realisation, documents through entire procurement process

Table 1 Requirements Characteristics

Following ISO 29148, ref. [ISO29148:2018] the characteristics for an overall specification are defined as:

Characteristics	Definition
Complete	• The set of requirements shall contain all necessary information
Consistent	• The set of requirements is free of conflicts to other requirements, OCORA specifications or standards, like ENs, and TSIs
Affordable	• The requirements can be satisfied by a solution that is obtainable / feasible within life cycle constraints
Bounded	• The set of requirements represent the scope needed only to satisfy the user needs
Unique	• The specification contains each requirement only once
Traceable	• The set of requirements has a relation to its goals and origin; the existence of the set of requirements can be comprehended
High-Quailty	• The set of requirements shall use correct lay-out, use correct English, be checked and be approved and all required meta data shall be available • All requirements in the specification fulfil the quality criteria for individual requirements

Table 2 Quality Criteria

3.2.2 Language Usage

To serve as a generic requirements catalogue, OCORA will publish all requirements in present tense.

The requirement classification will be exclusively managed using the dedicated requirements attribute “Requirement classification”. This means the requirement text remains always neutral regarding its classification.

Possible requirement classifications are:

- Requirement
- Design Recommendation
- Optional Requirement
- Information
- Operator Choice

In order to ensure that all requirements have the characteristics defined in Table 1, the terms in each requirement have to be chosen carefully. The following terms are to be avoided or to be used carefully:

- Comparatives and superlatives: (better, best, higher, most, smallest, largest)
- Subjective statements: (easy, good, user friendly, nice, good looking, inviting)
- Ambiguous terms: (always, optimal, minimal, maximum)
- Open ended statements: (at least, not limited to)
- Loopholes: (possible, as applicable)
- Negative statements: (has not, is not, no)
- Connective statements: (and, or)
- Passive voice (it is possible to)

4 Process

4.1 Quality Check

To illustrate how the theoretical approaches of chapter 3 can be checked in practice, a checklist for writing OCORA requirements and a OCORA quality metric to evaluate the quality of requirements is defined. This checklist is based on EuroSpec, ref. [EUROSPEC RM 2 0816].

This checklist is addressed to people who write OCORA requirements and shall help them to check their requirements by themselves.

1. Is the requirement formulated as a complete, clear and understandable sentence?
2. Is the requirement formulated in active voice and present time?
3. Is the requirement free of "weak words"?
4. Does the requirement describe a function or a property of a system or sub-system?
5. Does the requirement include only one requirement?
6. Is the requirement free of implicit assumptions?
7. Can the fulfilment of the requirement be verified?
8. Are all attributes filled in?

In addition, automatically generated reports in Polarion are supporting as additional quality checks.

4.2 Requirement Status

The requirement status is managed using the **Status attribute.** It may adopt the following values.

Value of attribute	Definition
Draft	A new work item automatically receives the status "Draft". The author works with the work item in this status until it is complete and ready for review.
In Review	The work item is ready for informal review. In this status, the work item can no longer be changed (except for status comments and approval status). Reviewers entered as Approvers in the Approvals attribute are invited to comment, approve or disapprove.
In Approval	The work item is ready for formal approval. In this status, the work item can no longer be changed (except for status and comments).
Approved	The work item is released. In this status, the work item can no longer be changed (except status).
Deleted	The work item is deleted. It remains in Polarion for documentation purposes, but is no longer considered by reports, for example.

Table 3 Requirement Status

The transition between the different status are described in the OCORA Methodology document, ref. [OCORA-BWS08-010] .

4.3 Requirements Approval Process

To monitor progress of approvals while drafting the status of the requirements is defined.

The status of requirements defines the development state of the requirements. Changing the status also changes the required activity of the requirement.

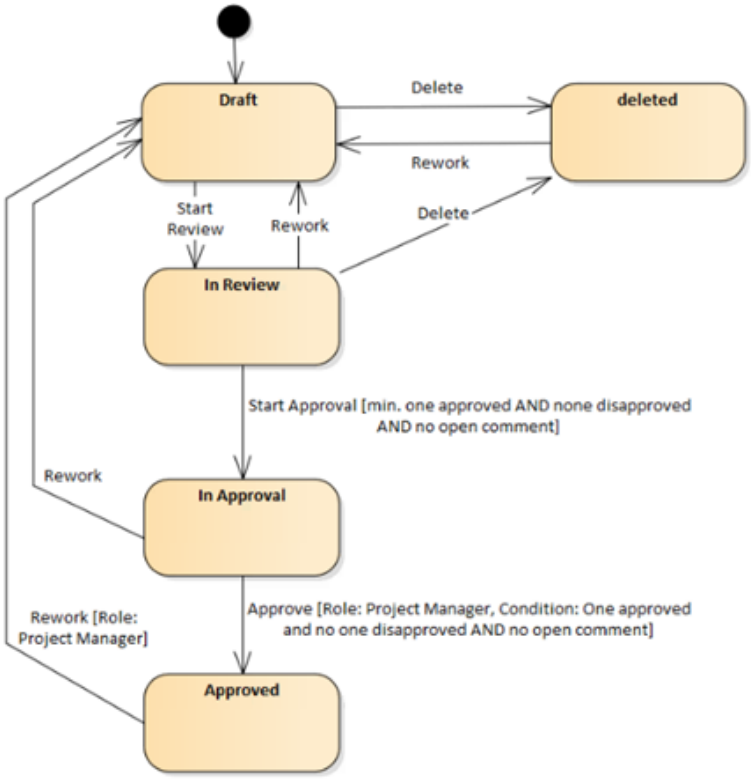

Figure 2 Status of Requirements

A requirement starts as a draft. Requirements which have to be reviewed will be set to the status “under review” and after the requirement has been approved, the status will be set accordingly. The status of requirements that are not approved will be set to draft or deleted. After all requirements have been approved, the OCORA Release as a whole can be approved, released and published.

4.4 Requirements Publishing Process

OCORA publishes content in subsequent releases.

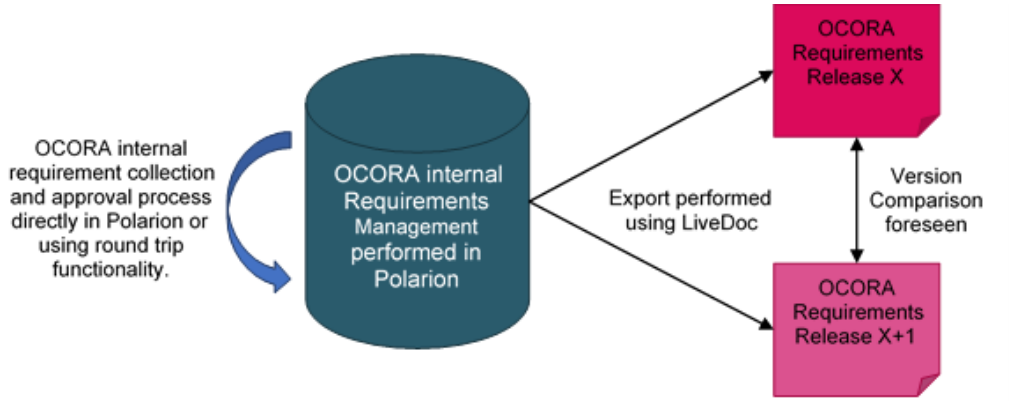

Figure 3 Requirements Publishing Process

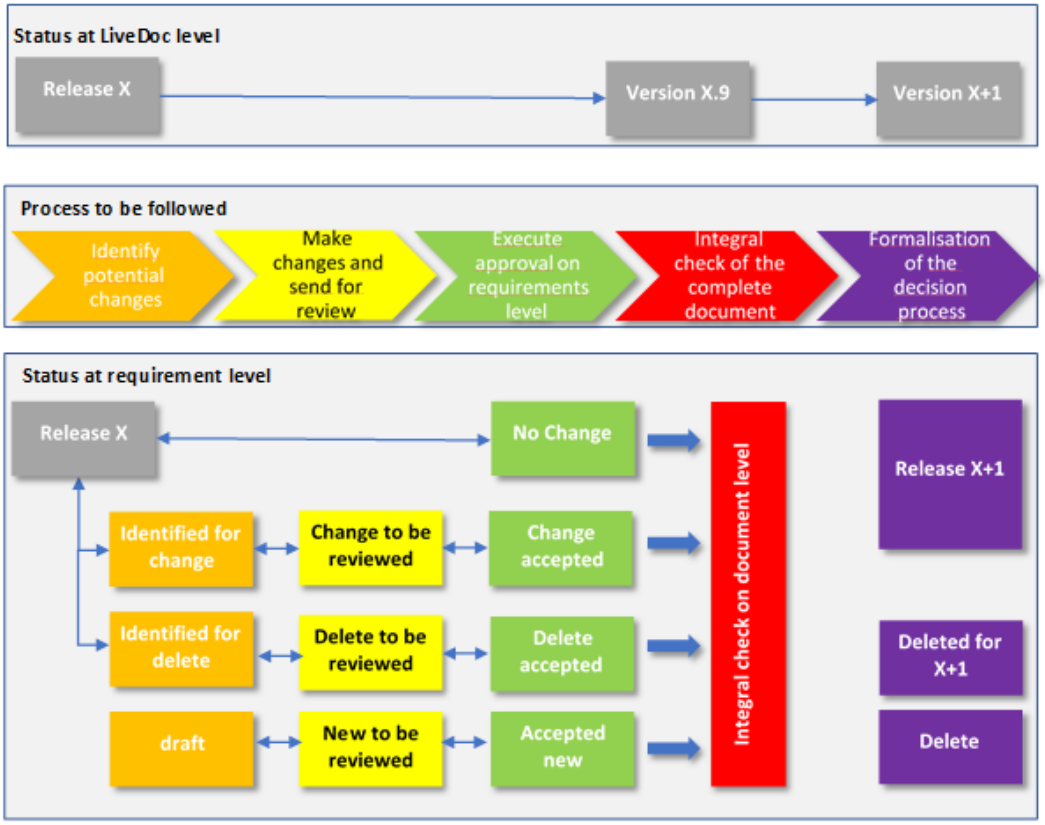

Figure 4 Baselining / Version Upgrade Process

5 Requirements Traceability

For the OCORA partners it is important to have a clear and common understanding of the term traceability and what it can be used for.

Traceability is the capability to comprehend and follow requirements, relationships, and dependencies between requirements during the whole life cycle of a system.

More specific traceability shall be done during the technical processes referring to the following V-model, based on [ISO15288:2015].

"Traceability covers at least two important aspects: the first aspect is traceability between various pieces of information at one point in time, for example traceability between customer requirements and system requirements. The second aspect is traceability of one single piece of information throughout time, for example how one requirement changed during the course of a project."

The first aspect of the ISO 152883 traceability is covered by this "Requirement Management Guideline" and the second aspect of the ISO 152883 traceability is covered by the "Baselining" process, see Figure 12.

Traceability is the foundation for efficient, high-quality requirements management, illustrating the relation of different artefacts. Traceability pursues the following objectives:

- Detectability: Requirement's traceability supports the proof, that requirements or objectives are implemented and fulfilled in a system.
- Impact analysis: Requirement's traceability supports impact analysis, for example by analysing the consequences for other requirements when changing a requirement (change management) or by analysing the influence between requirements.
- Identification of sophisticated requirements in specifications and system functions
- Different types of connections between requirements and information for traceability can be for example:
    - Relationship between requirements
    - Relationship between requirements and other artefacts, e.g. objectives, use-cases, test-cases, etc.

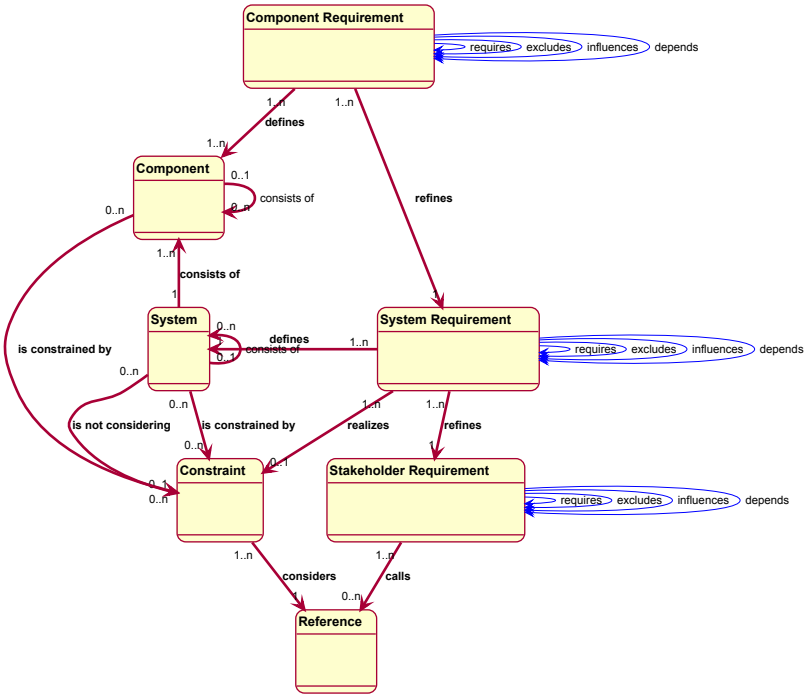

Figure 5 OCORA Requirement relations (linking)

6 Requirements Validation and Verification

For the OCORA it is important to have a clear and common understanding of validation and verification. The focus is to emphasize - from the train operating companies' point of view - what are the tasks related to validation and verification for the Railway Undertakings and how both can be implemented in the requirements specification during the process of purchasing rolling stock.

6.1 Common understanding of validation and verification

Validation and verification are important activities during all the technical processes of the V-Model and are important interfaces to requirements management and networked management disciplines like test management.

Validation and verification are used to demonstrate the requirements' compliance respectively by the tenderer, prior to contract signing and by the supplier, afterwards. Validation and verification is not only used for requirement (contractual) acceptance but also for risk management during the project.

Validation and verification are closely related but have the following main difference:

Validation is about checking whether a system meets the needs and expectations of the customer. Validation tests a system against the customer (or user) requirements.

Verification is about checking a system's conformity to its specification. Verification tests a system against system requirements and/or design requirements.

The difference between validation and verification can be pointed in the questions:

- Are we building the right CCS On-board system? (Validation)
- Are we building the CCS On-board system in a correct way? (Verification)

Validation and verification shall be planned as early as possible, as it influences test management. A proper preparation for the costly test-management is needed.

Validation and verification criteria's will be attached as linked work items to each requirement defined during the process of writing requirements once the OCORA Testing Strategy [OCORA-TWS09-010] is in place.

6.2 Methods for Validation and Verification

The following methods for validation and verification are applied based on EuroSpec, ref. [EUROSPEC RM 2 0816].

	Definition	Function	Applicability
Inspection	An inspection is an examination of the item against applicable documentation to confirm compliance with requirements.	Inspection is used to validate and verify properties best determined by examination and observation. Inspection is generally non-destructive and typically includes the use of sight, hearing, smell, touch, and taste; simple physical manipulation; mechanical and electrical gauging; and measurement.	Validation and Verification
1st Article Inspection (FAI)	A FAI is a special form of inspection of components, subsystems or systems manufactured under series conditions to see if it meets specifications and to ensure that the process can and does reliably produce what is intended. A FAI is carried out before approval of series production and is typically called for in a contract.	A FAI includes all relevant functional, non-functional, quality and produce process requirements to components, subsystems or systems. A FAI can be carried out when all corresponding validation and verification methods - for example design reviews, type tests, calculations etc.- are successfully finished.	Quality Management (results in a Project Management Gate)
Analysis	Use of analytical data or simulations under defined conditions to show theoretical compliance	Used where testing to realistic conditions cannot be achieved or is not cost effective. Analysis may be used when such means establish that the appropriate requirement, specification or derived requirement is met by the proposed solution.	Testing Validation
Design Review	A formal and systematic validation within a product development process of components, subsystems or systems whereby a design is detailed and evaluated against its requirements.	Validation of agreed functional and non-functional requirements of the system and identification and elimination of potential problems and errors as early as possible.	Testing Verification Validation
Simulation	Simulation is the imitation of the operation of a real-world process or system over time. A model represents the key characteristics, behaviors or functions of the selected physical or abstract system or process. The model represents the system itself, whereas the simulation represents the operation of the system over time. Simulations can be used to show, under defined conditions, theoretical compliance.	Used where testing to realistic conditions cannot be achieved or is not cost effective. Simulation may be used when such means establish that the appropriate requirement, specification, or derived requirement is met by the proposed solution. Simulation can be used e.g. in the process of development to show movement and functional processes, dynamic loads and stresses.	Testing

	Definition	Function	Applicability
Calculation	Calculation to validate or verify agreed or specified parameter, under defined parameters and rules of calculation, by a mathematical proof to show theoretical compliance.	For the input-parameters of the calculation, methods of calculation and the result parameter are presented and evaluated.	Testing Validation
Demonstration	A qualitative exhibition of functional performance usually accomplished with no or minimal instrumentation or test equipment. Demonstration uses a set of test activities with system stimuli selected by the supplier to show that system or system element response to stimuli is suitable or to show that operators can perform their allocated functions when using the system. Observations are made and compared with predetermined responses.	Demonstration may be appropriate when requirements or specifications are given in statistical terms (e.g., mean time to repair, average power consumption, etc.).	
Routine Test	Examination of every component, subsystem or system during or after the manufacturing process to prove its compliance with the requirements.		
Test	An action by which the operability, maintainability or performance capability of an item is quantitatively verified when subjected to controlled conditions that are real or simulated.	These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis.	Testing
Type test	Test of one or more components, subsystems or the system to prove that the construction is in compliance with the required specification and the relevant standards.	The test object does not need to be manufactured under series conditions. Type testing includes the validation of required parameter. Several type tests can be necessary for one test object.	
Certification	Certification is a written assurance that the system or system element has been developed in accordance with the required standard and meets the requirements. This assures that the system or system element can perform its assigned functions to a negotiated standard.	The development reviews and system verification and validation results form the basis for certification. Certification is generally performed by a third party against an accepted standard.	

Table 4 Validation & Verification

7 Polarion Work Item Types

7.1 Overview

The following class diagram provides an overview of the work items and their linkage. The light-yellow boxes represent Polarion work items, the blue boxes illustrate LiveDocs and the green boxes represent containers for an entire group of Polarion work items which are part of the OCORA MBSE Polarion project. The OCORA MBSE project will contains Capella artefacts exported from the OCORA Capella MBSE Model.

Yellow bubbles are notes with further explanations; and green bubbles indicate where the requirements of the four distinct OCORA Requirement Levels can be found.

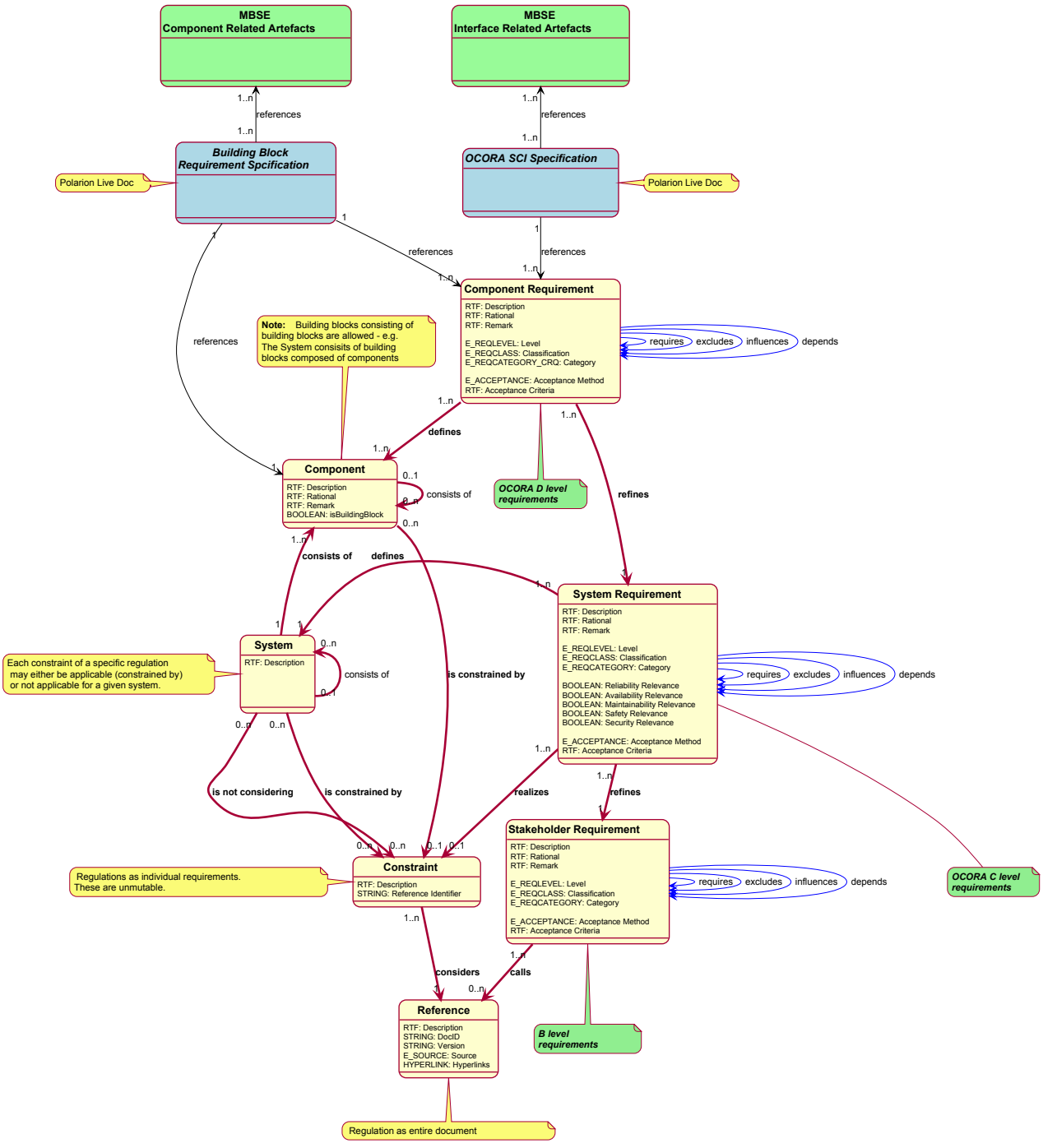

Figure *6* Work Item Type Overview

7.2 Work Item Type: Reference

The Reference work item type is used to refer to a source (document) outside Polarion. Attributes include the source's (document's) meta data like name and version but as well as the possibility to link an external source via URL.

Note: Ideally the source document is being attached to the Reference and not linked via an external URL.

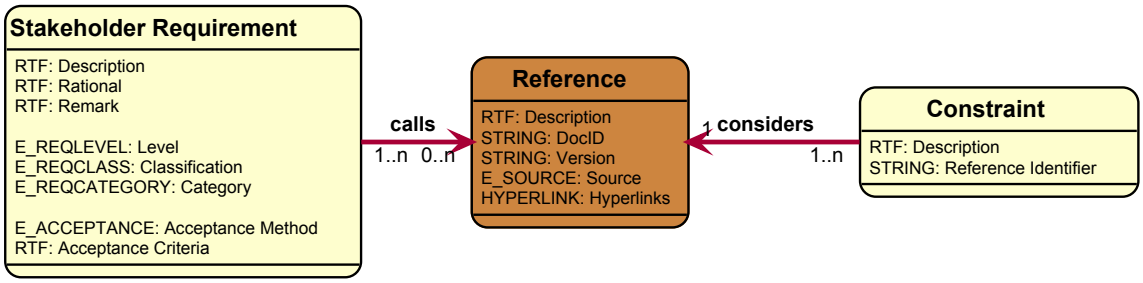

Figure 7 Work Item *Reference*

7.2.1 Attributes

Name	Type	Mandatory	Description
Title	String	yes	Title of the source (document)
Description	Rich Text	no	Short description of source (document content)
DocID	String	yes	Unique identifier of the source (document). The format of the ID is source specific.
Version	String	yes	Identifies the exact version of the source (document)
Source	ENUM	yes	Identifies the type of source (document) regarding where it comes from e.g. internal or external
Attachment	File	no(*)	A copy of the source (document) in case it is not stored on an external repository (see hyperlinks).
Hyperlinks	url	no(*)	URL link to the referenced source (document). Note: the danger of using links to external repositories is that they may become obsolete over time. Attaching the file should be the preferred approach.

Table 5 Reference Attributes

(*) One of the two is mandatory - the author may choose which one to use.

7.2.2 Outgoing Links

No outgoing links are allowed for the Reference work item type.

7.3 Work Item Type: Constraint

The Constraint work item type represents a single constraint of a referenced source (documents like norms, standards, regulations, etc.). The description attribute corresponds to an exact copy of an unambiguously identifiable text block in the considered Reference (document).

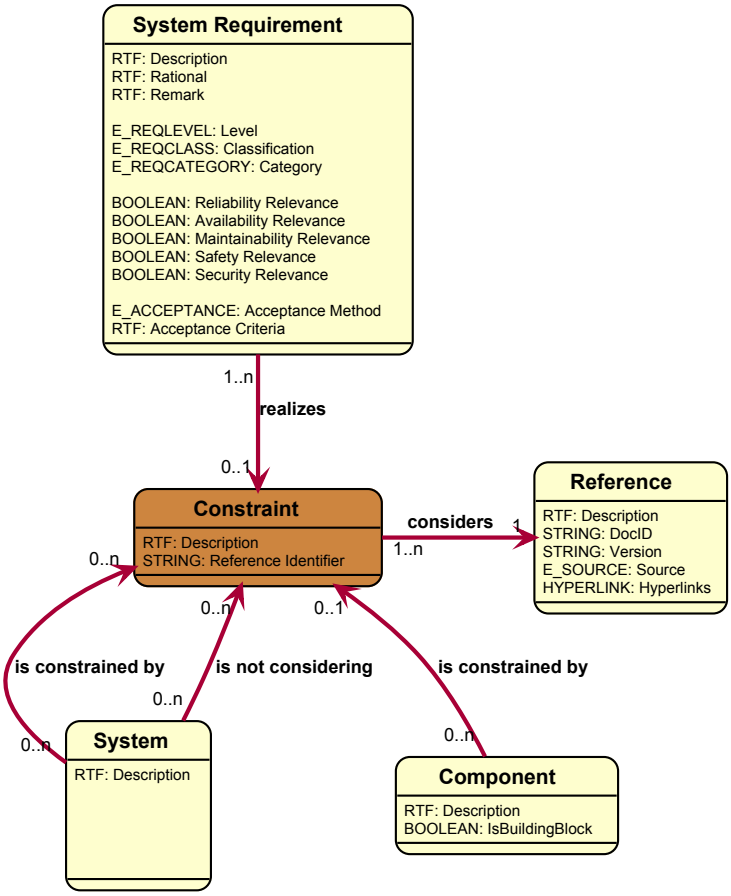

Figure *8* Work Item *Constraint*

7.3.1 Attributes

Name	Type	Mandatory	Description
Title	String	yes	Title of the constraint, summarizing in a few words the content of the constraint
Description	Rich Text	yes	Exact copy of a uniquely identifiable text block out of the considered *Reference* (document)
Reference Identifier	String	yes	Unique identifier pointing to the text block in the considered *Reference* (document). The format of the *Reference Identifier* is *Reference* (document) specific.
Linked Work Items	link role	no	List of links to other work items as per the out going link roles (see below)

Table 6 Constraint Attributes

7.3.2 Out going Links

Link role	Inverse role	To	Multiplicity	Description
considers	is considered by	*Reference*	1	Link to *Reference* (document) e.g., norm, standard, regulation, etc. from where the *Constraint* originates.

Table 7 Constraint out going links

7.4 Work Item Type: System

The System work item represents a system or a sub-system. A system is considered a sub-system in case another system has a “consists of” link to the respective system.

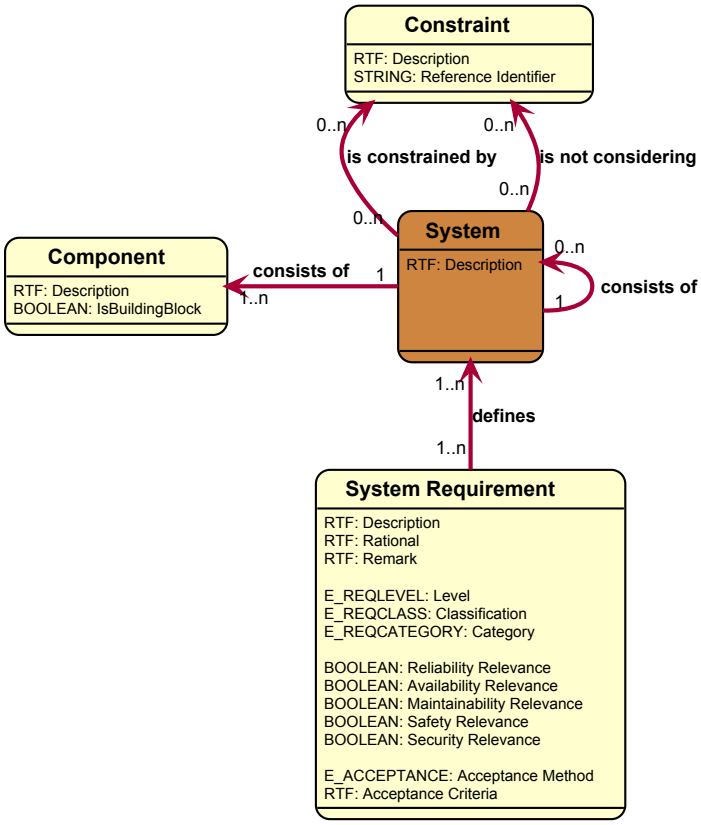

Figure 9 Work Item *System*

7.4.1 Attributes

Name	Type	Mandatory	Description
Title	String	yes	Name of the system
Description	Rich Text	no	A short description of the system
Linked Work Items	link role	no	List of links to other work items as per the out going link roles (see below)

Table 8 System Attributes

7.4.2 Out going Links

Link role	Inverse role	To	Multiplicity	Description
consists of	is part of	*Component*	0..*	Links the *Components* that the *System* is composed of.
is constrained by	constrains	*Constraint*	0..*	Link to Constraints that are applicable to the system. **Note:** to ensure that all *Constraints* for a given *Reference* (document) have been evaluated, each *Constraint* must either be linked via this link role or by the *is not considering* link role.
is not considering	is not considered by	*Constraint*	0..*	Link to Constraints that are applicable to the system. **Note:** to ensure that all *Constraints* for a given *Reference* (document) have been evaluated, each *Constraint* must either be linked via this link role or by the *is constrained by* link role.
consists of	is part of	*System*	0..*	Identifies a system as being part of this system e.g., as being a sub-system of the system.

Table 9 System out going links

7.5 Work Item Type: Component

This work item represents a component of a system (or sub-system). Components may consist of other components and may be considered an OCORA building block. Building Block are separately source-able components that may be tendered independently. Each building block has a dedicated Building Block Requirement Specification in form of a Polarion LiveDoc.

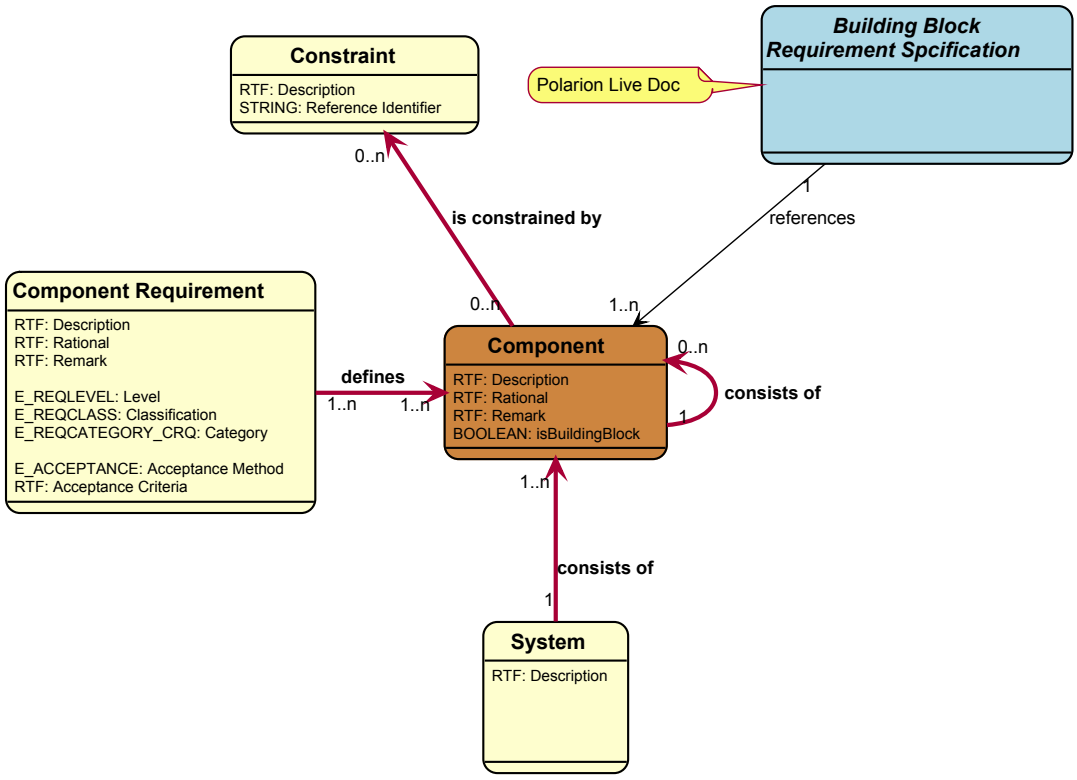

Figure *10* Work Item *Component*

7.5.1 Attributes

Name	Type	Mandatory	Description
Title	String	yes	Name of the component
Description	Rich Text	no	A short description of the component
IsBuildingBlock	Boolean	no	Identifies a component as an OCORA building block (a separately source-able component)
Linked Work Items	link role	no	List of links to other work items as per the out going link roles (see below)

Table 10 Component Attributes

7.5.2 Out going links

Link role	Inverse role	To	Multiplicity	Description
is defined by	defines	*Component Requirement*	0..*	Links to all *Component Requirements* that define this component.
is constrained by	constrains	*Constraint*	0..*	This link is used for a high-level assignment of constraints to components. **Note:** this is only to be used to allow for a rough assignment. The detailed traceability path will always be via *Constraint -> System Requirement -> Component Requirement -> Component.*
consists of	is part of	*Component*	0..*	Links the *Components* that the *System* is composed of.

Table 11 Component out going links

7.6 Work Item Type: Stakeholder Requirement

The stakeholder requirement work item type is used for all OCORA A-Level and B-Level requirements. They represent the initial formalized requirements of OCORA. All requirements must be verifiable either via tests or any other mean. Stakeholder requirements are the first formal elements to be refined in further steps, deriving required OCORA Building Blocks.

Stakeholder requirements:

- define the OCORA scope
- must be comprehensible to the stakeholder(s) of the overall system
- must be complete regarding the scope
- include all high-level requirements (functional & non-functional) of the overall system
- will be systematically refined in further steps (system requirements / component requirements)

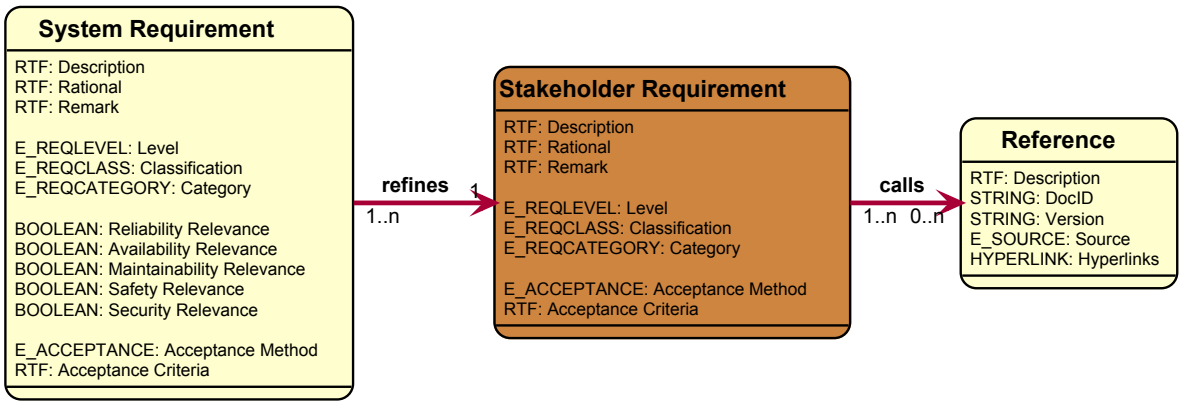

Figure *11* Work Item *Stakeholder Requirement*

7.6.1 Attributes

Name	Type	Mandatory	Description
Title	String	yes	Describes in a short sentence the essence of the requirement
Description	Rich Text	yes	The formal description of the requirement. **Note:** The formal description of the requirement must be written in a pure factual way using the present tense. The necessity of the requirement is exclusively determined using the *Classification* attribute.
Rational	Rich Text	yes	Provides additional information to the reader regarding the reason why this is a requirement of the overall system. It shall assist taking the right refinement decisions.
Remark	Rich Text	no	Information that can neither be included in the formal *Description* nor in the *Rational*.
Level	ENUM	yes	Identifies the requirement level as per the OCORA requirement structure (3.1 - Requirements Structure). *Possible values are*: • A-Level • B-Level
Classification	ENUM	yes	Defines the necessity of the requirement. **Note:** the necessity of the requirement is exclusively determined by this attribute. The formal description of the requirement must be written in a pure factual way using the present tense. *Possible values are*: • Requirement • Optional Requirement • Design Recommendation • Operator Choice • Information
Category	ENUM	yes	For stakeholder requirements there is only a differentiation between functional and non-functional requirements. Further refinement regarding the type of non-functional requirement (RAMSS), will be introduced with the system requirement. *Possible values are*: • Functional • Non-functional

Name	Type	Mandatory	Description
Acceptance Method	ENUM	yes	Defines the method to be used to verify compliance of a solution with this requirement. *Possible values are*: • Analysis • Calculation • Certification • Demonstration • Design Review • Inspection • Process Review • Simulation • Test
Acceptance Criteria	Rich Text	no	Describes the criteria to be met to accept compliance with this requirement.
Linked Work Items	link role	no	List of links to other work items as per the out going link roles (see below)

Table 12 Stakeholder Requirement Attributes

7.6.2 Out going links

Link role	Inverse role	To	Multiplicity	Description
calls	is called by	*Reference*	0..*	Reference to an external source (document) e.g., norm, standard, regulation, etc.
requires	is required by	*Stakeholder Requirement*	0..*	R1 requires R2: This relationship implies that R2 is needed to fulfil R1, i.e. R2 is a pre-condition for R1.
excludes	is excluded by	*Stakeholder Requirement*	0..*	R1 excludes R2: This means that R1 and R2 are alternatives and only one of them can be selected.
influences	is influenced by	*Stakeholder Requirement*	0..*	R1 influences R2: This relationship means that the inclusion of R1 in the requirements specification causes a change in the cost or in the priority of R2. For instance, the inclusion of R1 implies that the cost of R2 decreases in 2 (value) persons-months (valueUnit).
depends on	is dependent on	*Stakeholder Requirement*	0..*	R1 depends on R2: This means that there exists a relationship between R1 and R2 that is neither requires, nor excludes, nor influences. It is just the way to explicitly show that R1 is related to R2.

Table 13 Stakeholder Requirement out going links

7.7 Work Item Type: System Requirement

The system requirement work item type is used for all OCORA C-Level requirements. They represent the full set of formalized requirements of system that have been derived/refined directly or indirectly from the stakeholder requirements. Indirectly means via constraints (see 4.3 Work Item Type: Constraint) originating from Standards and Regulations referenced in stakeholder requirements. All requirements must be verifiable either via tests or any other mean. System requirements are further refined in component requirements, during the architecting process.

System requirements:

- define the scope of the system that is needed to comply with the stakeholder requirements.
- must be comprehensible to the architect(s) of the system
- must be complete regarding the scope
- include all requirements (functional & non-functional) of the system
- will be systematically refined during the architecting process (component requirements)

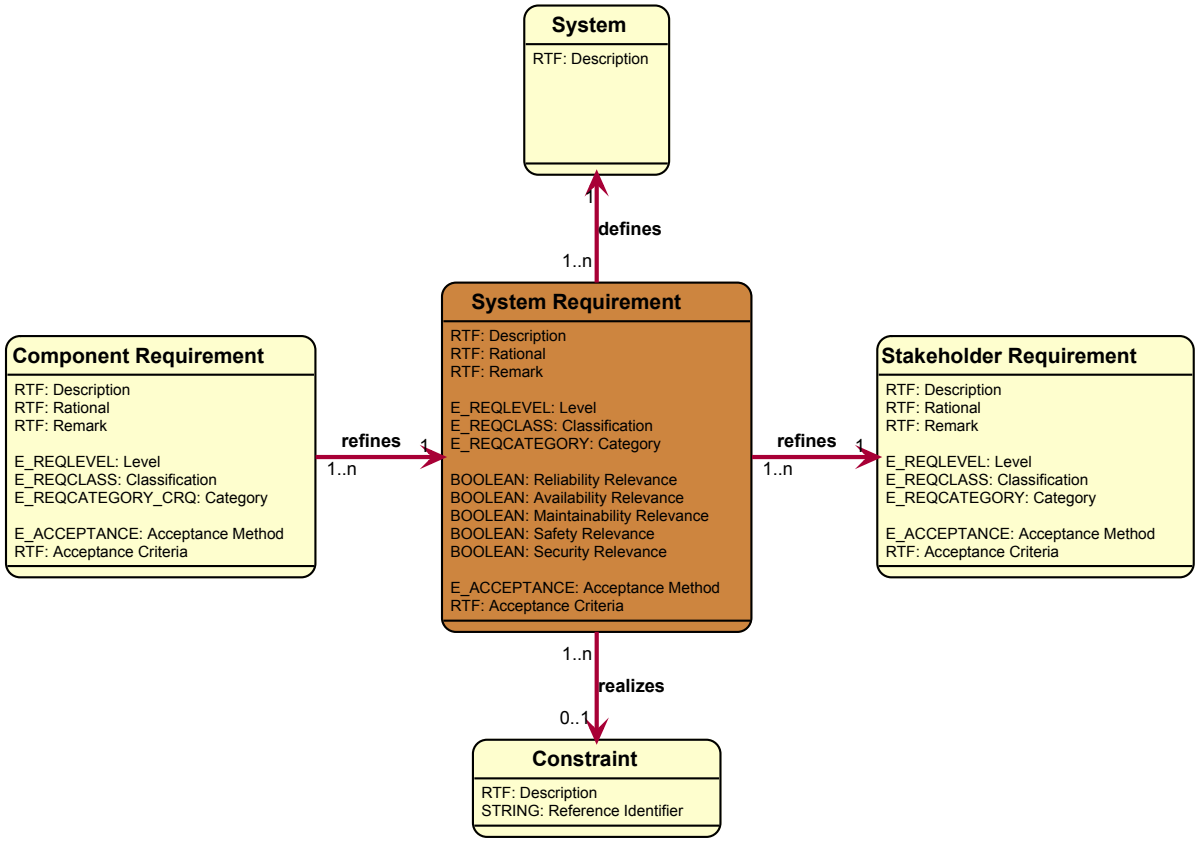

Figure *12* Work Item *System Requirement*

7.7.1 Attributes

Name	Type	Mandatory	Description
Title	String	yes	Describes in a short sentence the essence of the requirement
Description	Rich Text	yes	The formal description of the requirement. **Note:** The formal description of the requirement must be written in a pure factual way using the present tense. The necessity of the requirement is exclusively determined using the *Classification* attribute.
Rational	Rich Text	yes	Provides additional information to the reader regarding the reason why this is a requirement of the overall system. It shall assist taking the right refinement decisions.
Remark	Rich Text	no	Information that can neither be included in the formal *Description* nor in the *Rational*.
Level	ENUM	yes	Identifies the requirement level as per the OCORA requirement structure (3.1 - Requirements Structure). *Possible values are*: • C-Level
Classification	ENUM	yes	Defines the necessity of the requirement. **Note:** the necessity of the requirement is exclusively determined by this attribute. The formal description of the requirement must be written in a pure factual way using the present tense. *Possible values are*: • Requirement • Optional Requirement • Design Recommendation • Operator Choice • Information
Category	ENUM	yes	System requirements are on a high-level either functional or non-functional. Their relevance regarding RAMSS is specified using dedicated attributes. *Possible values are*: • Functional • Non-functional
Reliability Relevance	Boolean	no	Indicates that this requirement influences the reliability or is influenced by the reliability.
Availability relevance	Boolean	no	Indicates that this requirement influences the availability or is influenced by the availability

Name	Type	Mandatory	Description
Maintainability relevance	Boolean	no	Indicates that this requirement influences the maintainability or is influenced by the maintainability
Safety Relevance	Boolean	yes	Indicates that this requirement influences the safety or is influenced by the safety
Security Relevance	Boolean		Indicates that this requirement influences the security or is influenced by the security
Acceptance Method	ENUM	yes	Defines the method to be used to verify compliance of a solution with this requirement. *Possible values are*: • Analysis • Calculation • Certification • Demonstration • Design Review • Inspection • Process Review • Simulation • Test
Acceptance Criteria	Rich Text	no	Describes the criteria to be met to accept compliance with this requirement.
Linked Work Items	link role	no	List of links to other work items as per the out going link roles (see below)

Table 14 System Requirement Attributes

7.7.2 Out going links

Link role	Inverse role	To	Multiplicity	Description
defines	is defined by	*System*	1	Link to the system that this requirement belongs to
realizes	is realized by	*Constraint*	0..1	Link to a constraint that the requirement realizes.
refines	is refined by	*Stakeholder Requirement*	1	Link to the Stakeholder Requirement that this requirement has been derived/refined from.

Link role	Inverse role	To	Multiplicity	Description
requires	is required by	*System Requirement*	0..*	R1 requires R2: This relationship implies that R2 is needed to fulfil R1, i.e. R2 is a pre-condition for R1.
excludes	is excluded by	*System Requirement*	0..*	R1 excludes R2: This means that R1 and R2 are alternatives and only one of them can be selected.
influences	is influenced by	*System Requirement*	0..*	R1 influences R2: This relationship means that the inclusion of R1 in the requirements specification causes a change in the cost or in the priority of R2. For instance, the inclusion of R1 implies that the cost of R2 decreases in 2 (value) persons-months (valueUnit).
depends on	is dependent on	*System Requirement*	0..*	R1 depends on R2: This means that there exists a relationship between R1 and R2 that is neither requires, nor excludes, nor influences. It is just the way to explicitly show that R1 is related to R2.

Table 15 System Requirement out going links

7.8 Work Item Type: Component Requirement

The component requirement work item type is used for all OCORA D-Level requirements. They represent the full set of formalized requirements of all components of the system. They are derived/refined from the overall system requirements during the architecting process. All requirements must be verifiable either via tests or any other mean.

Component requirements:

- define the scope of all components that the system comprises
- must be comprehensible to future component architect(s)
- must be complete regarding the component scope
- include all requirements (functional & non-functional) of the components
- include all requirements regarding standardised interfaces between components

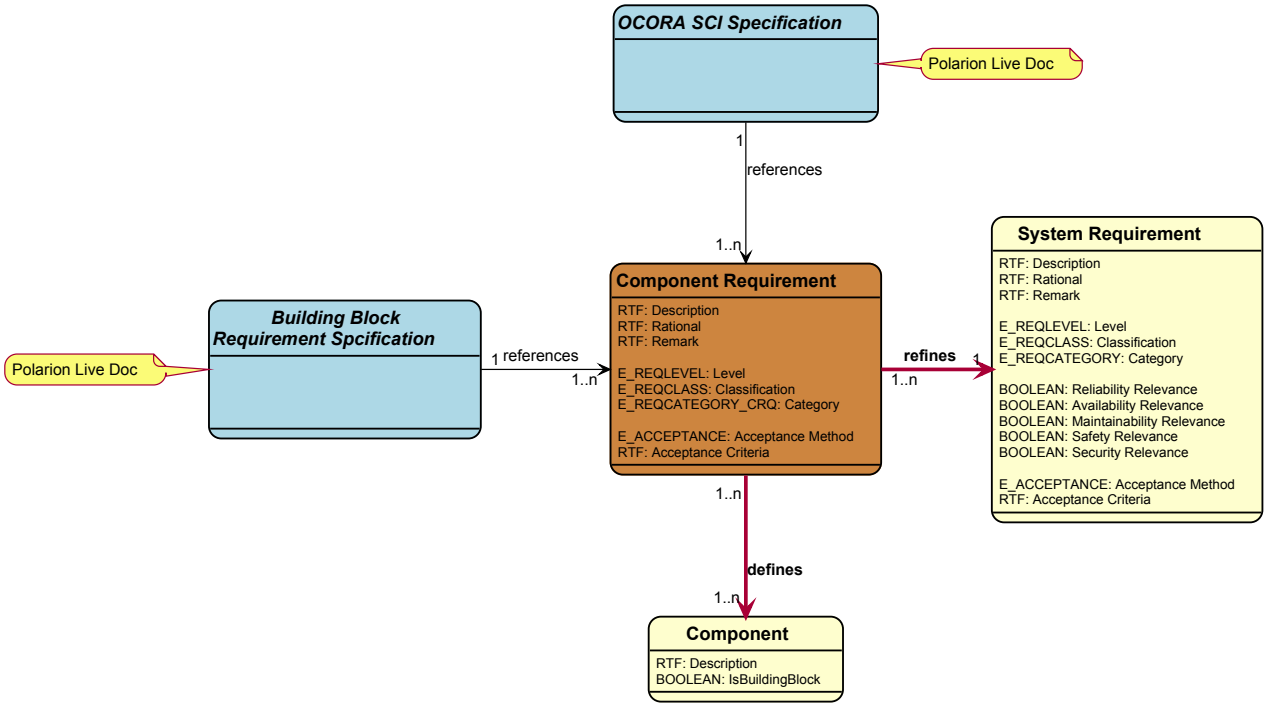

Figure *13* Work Item *Component Requirement*

7.8.1 Attributes

Name	Type	Mandatory	Description
Title	String	yes	Describes in a short sentence the essence of the requirement
Description	Rich Text	yes	The formal description of the requirement. **Note:** The formal description of the requirement must be written in a pure factual way using the present tense. The necessity of the requirement is exclusively determined using the *Classification* attribute.
Rationale	Rich Text	yes	Provides additional information to the reader regarding the reason why this is a requirement of the overall system. It shall assist taking the right refinement decisions.

Name	Type	Mandatory	Description
Remark	Rich Text	no	Information that can neither be included in the formal *Description* nor in the *Rational*.
Level	ENUM	yes	Identifies the requirement level as per the OCORA requirement structure (3.1 - Requirements Structure). *Possible values are*: • D-Level
Classification	ENUM	yes	Defines the necessity of the requirement. **Note:** the necessity of the requirement is exclusively determined by this attribute. The formal description of the requirement must be written in a pure factual way using the present tense. *Possible values are*: • Requirement • Optional Requirement • Design Recommendation • Operator Choice • Information
Category	ENUM	yes	Component requirements must be refined in such a way, that they can be assigned to exactly one single category. *Possible values are*: • Functional • Reliability • Availability • Maintainability • Safety • Security • NFR non-RAMSS
Acceptance Method	ENUM	yes	Defines the method to be used to verify compliance of a solution with this requirement. *Possible values are*: • Analysis • Calculation • Certification • Demonstration • Design Review • Inspection • Process Review • Simulation • Test
Acceptance Criteria	Rich Text	no	Describes the criteria to be met to accept compliance with this requirement.
Linked Work Items	link role	no	List of links to other work items as per the out going link roles (see below)

Table 16 Component Requirement Attributes

7.8.2 Out going links

Link role	Inverse role	To	Multiplicity	Description
defines	is defined by	*Component*	1	Link to the component that this requirement belongs to
refines	is refined by	*System Requirement*	1	Link to the System Requirement that this requirement has been derived/refined from.
requires	is required by	*Component Requirement*	0..*	R1 requires R2: This relationship implies that R2 is needed to fulfil R1, i.e. R2 is a pre-condition for R1.
excludes	is excluded by	*Component Requirement*	0..*	R1 excludes R2: This means that R1 and R2 are alternatives and only one of them can be selected.
influences	is influenced by	*Component Requirement*	0..*	R1 influences R2: This relationship means that the inclusion of R1 in the requirements specification causes a change in the cost or in the priority of R2. For instance, the inclusion of R1 implies that the cost of R2 decreases in 2 (value) persons-months (valueUnit).
depends on	is dependent on	*Component Requirement*	0..*	R1 depends on R2: This means that there exists a relationship between R1 and R2 that is neither requires, nor excludes, nor influences. It is just the way to explicitly show that R1 is related to R2.

Table 17 Component Requirement out going links

7.9 Mapping of OCORA and EuroSpec attributes

OCORA Attribute	EuroSpec Attribute	EuroSpec Definition
Unique Identification	ID	Unique Identification of the requirement
Status	Status	Indication of the approval state of the requirement
Category	Requirement type	Assignment of requirements into defined groups
Classification	Requirement classification	Importance and legal status of the requirement to the project
Description	Requirement-text	Description of the requirement
Rationale	Rationale	Reason to state the requirement
Foreseen by OCORA as linked work item, (e.g. TSI Subset)	Source	Indication from where the requirement originates
Owner	Owner	Owner of the requirement
Not foreseen by OCORA, can be added by tendering party later	Degree of fulfilment	Expected measure of compliance by the tender
Linked Work Items and Hyperlinks	Annex to requirement	Any appendices to the requirement given to the supplier industry
Comments	Comment of Requirement Review Board (RRB)	Remarks of the RRB to the requirement
Acceptance Criteria **Please note:** Test Cases will be defined as Polarion work items “Test Case” and will be linked to the respective requirements in the attribute Acceptance Criteria.	Verification (points in time)	Indication of methods and/ or documentation to verify a requirement

Table 18 Requirement Attribute mapping

The following EuroSpec attributes according to [EUROSPEC RM 2 0816] do not have a 1:1 mapping to OCORA attributes: Version, Requirement type, Decisions, Comment of Owner, Product element EN 15380-2, Function element EN 15380-4, Change since last release.