 THE RESEARCH LEADING TO THESE RESULTS HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S SEVENTH FRAMEWORK PROGRAM (FP7/2007-2013) FOR CRYSTAL – CRITICAL SYSTEM ENGINEERING ACCELERATION JOINT UNDERTAKING UNDER GRANT AGREEMENT N° 332830 AND FROM SPECIFIC NATIONAL PROGRAMS AND / OR FUNDING AUTHORITIES.

CRitical S**YST**em Engineering **A**cce**L**eration

MSE Report – V1

D203.011

DOCUMENT INFORMATION

Project	CRYSTAL
Grant Agreement No.	ARTEMIS-2012-1-332830
Deliverable Title	MSE Report – V1
Deliverable No.	D203.011
Dissemination Level	CO
Nature	R
Document Version	V1.00
Date	2014-01-31
Contact	Ralf BOGUSCH
Organization	EADS-CAS
Phone	+ 49 7545.8-2745
E-Mail	ralf.bogusch@cassidian.com

AUTHORS TABLE

Name	Company	E-Mail
Sascha ACKVA	EADS-CAS	sascha.ackva@cassidian.com
Inga BINDER	EADS-CAS	inga.binder@cassidian.com
Ralf BOGUSCH	EADS-CAS	ralf.bogusch@cassidian.com
Michael LEUTE	EADS-CAS	michael.leute@cassidian.com
Robert WÖHLER	EADS-CAS	robert.woehler@cassidian.com

REVIEW TABLE

Version	Date	Reviewer
V0.10	2013-12-17	Uwe KÜHNE (EADS-CAS), Tobias SORG (EADS-CAS)
V0.20	2014-01-17	Frédéric AUTRAN (EADS-CAS), Adeline SCHÄFER (FhG), Herbert KLENK (EADS-CAS), Marc MALOT (SAGEM), Andreas MITSCHKE (EADS IW-G)

CHANGE HISTORY

Version	Date	Reason for Change	Pages Affected
V0.10	2013-12-17	Initial Version for internal review	All
V0.20	2014-01-17	Version for external review	All
V1.00	2014-01-31	Version for release	

CONTENT

List of Figures

List of Tables

1 Introduction

1.1 Role of the Deliverable

This document has the following major purposes:

- Define the overall use case, including a detailed description of the underlying development processes and the set of involved process activities and engineering methods
- Provide input to WP601 (IOS Development) required to derive specific IOS-related requirements
- Provide input to WP602 (Platform Builder) required to derive adequate meta models
- Establish the technology baseline with respect to the use case, and the expected progress beyond (existing functionalities vs. functionalities that are expected to be developed in CRYSTAL)

1.2 Relationship to other CRYSTAL Documents

This document is the first in a series of four reports:

- D203.011 – MSE Report V1: use case definition (this document)
- D203.012 – MSE Report V2: update of MSE Report V1 and evaluation results from first iteration
- D203.013 – MSE Report V3: update of MSE Report V2 and evaluation results from second iteration
- D203.014 – Final MSE Report: update of MSE Report V3

This document provides input to the following CRYSTAL deliverables:

- D203.020 – First MSE SEE (Prototype)
- D601.021 – Interoperability Specification V1
- D602.011 – Meta-model Specification V1
- D604.011 – Specification, Development and Assessment for Safety Engineering V1
- D607.011 – Specification, Development and Assessment for Requirements-based Engineering V1
- D608.011 – Specification, Development and Assessment for Product Lifecycle Management V1
- D610.011 – Crystal Variability Management V1
- D610.031 – System Family Engineering Framework V1
- D611.011 – Specification, Development and Assessment for Software Development Lifecycle Management V1

1.3 Structure of this Document

This document is composed of three main chapters:

- Chapter 1 gives an overview of the scope of the deliverable, relationship with other CRYSTAL documents and this description of the document structure.
- Chapter 2 introduces the Mission Support Equipment (MSE) family to which this use case applies.
- Chapter 3 describes in detail the development processes and the engineering methods relevant for this use case. The information is generated from a SysML model that has been setup to support the definition of the use case.
- Annex I depicts the current technology baseline and identifies the gaps that need to be addressed by CRYSTAL technical innovations and technology bricks.
- Annex II provides the detailed engineering method forms.
- Annex III defines the requirements that shall be fulfilled by the CRYSTAL Reference Technology Platform (RTP), Interoperability Specification (IOS) and the technology bricks in order to enable building a System Engineering Environment (SEE) with ready-for use industrial tool chains.

1.4 How to Read this Document

In order to support the IOS Needs Capturing Process that shall be performed by every use case work package, an IBM Rational Rhapsody model has been setup that describes the selected User Stories, Process Activities, Engineering Methods, bricks, generic tool functions and requirements to be fulfilled by the CRYSTAL SEE. The model-based approach is based on SysML with additional stereotypes representing the model elements used in the IOS Needs Capturing Process. Figure 1-1 gives an overview of the model-based approach.

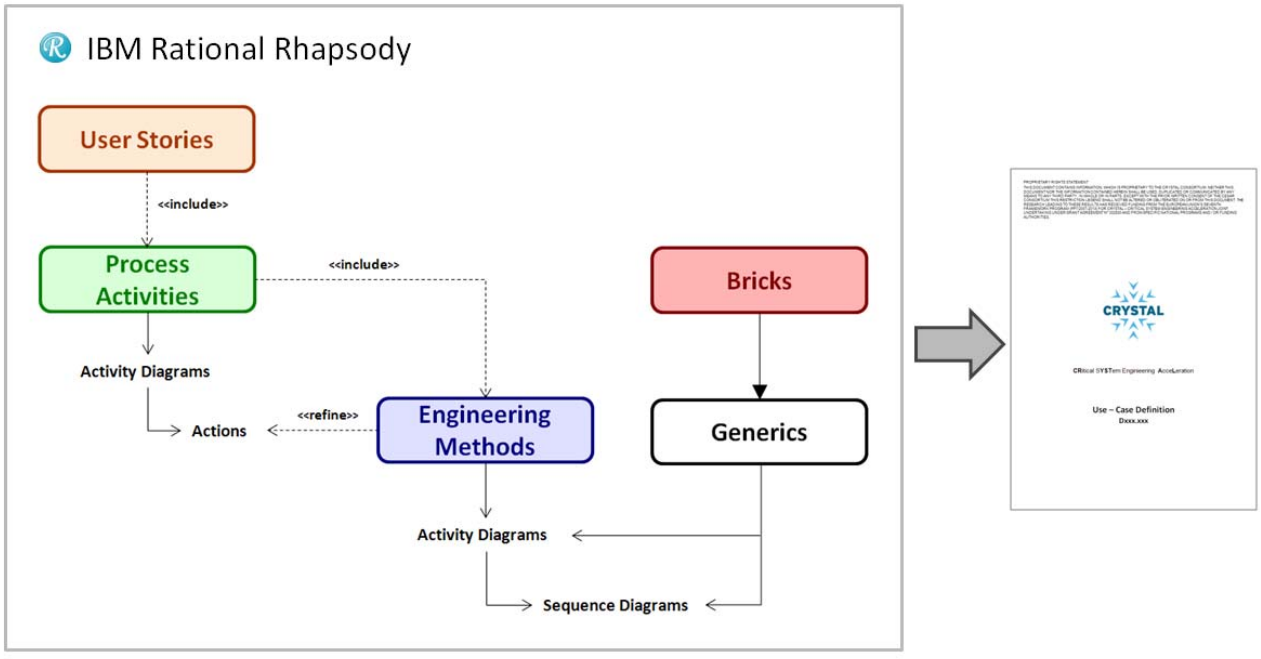

Figure 1-1 Model-based Approach Supporting the IOS Needs Capturing Process

In the following additional information is provided in order to support the reader not familiar with the adopted model-based approach.

The structure of Chapter 3 is based on five User Stories (US) selected for this use case:

- US202 - Safety analyses,
- US203 - Variability management,
- US204 - Ontology-based requirements engineering,
- US205 - Process automation, guidance and monitoring,
- US206 - Project compliance monitoring based on advanced traceability.

These User Stories have been introduced in the [CRYSTAL DOW]. A use case diagram presents all selected user stories and the involved CRYSTAL partners.

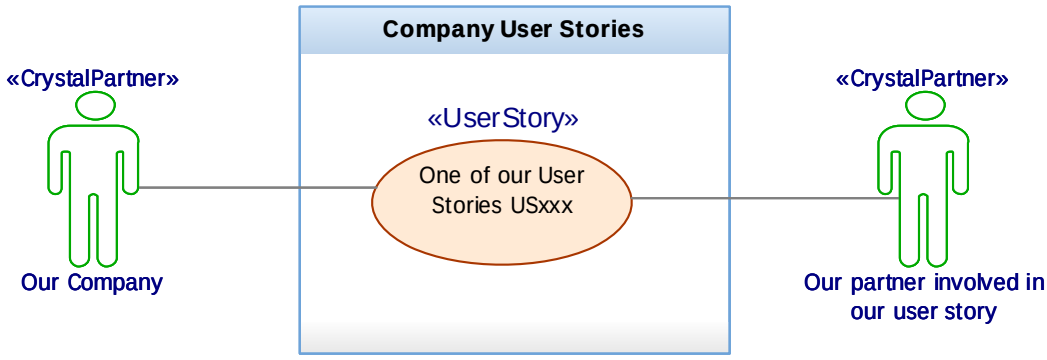

Figure 1-2 Use Case Diagram – Overview of User Stories

Each User Story includes one or more Process Activities (PA). A Process Activity is a detailed description of the use case process. It describes the process as a set of activities from the Cassidian perspective. The dependency of the Process Activities to the User Story is described in a use case diagram, see Figure 1-2.

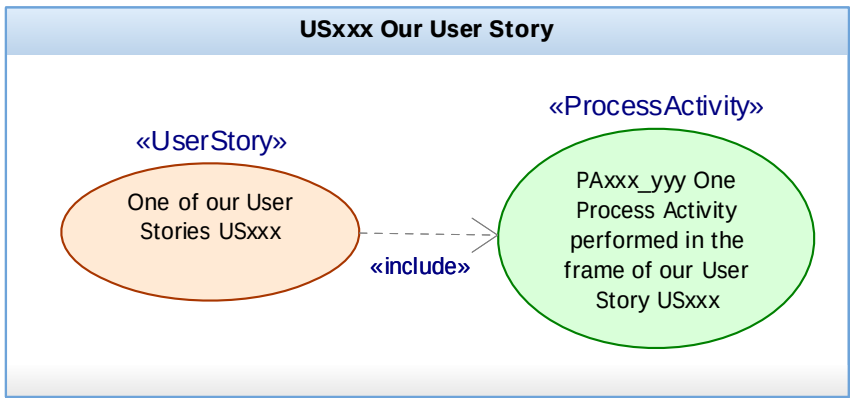

Figure 1-3 Use Case Diagram – Overview of Process Activities

Each Process Activity is described in more detail by an activity diagram. In order to cope with complexity, an activity diagram may be decomposed into several sub-activity diagrams. The activity diagram represents the

functional flow of the Process Activity using actions, control flows, forks, joins, decision and merge nodes. Within the Process Activity, Engineering Methods (EM) are identified by assigning the stereotype <<RelatesToEngMethod>> to the respective action of the activity diagram, see Figure 1-3.

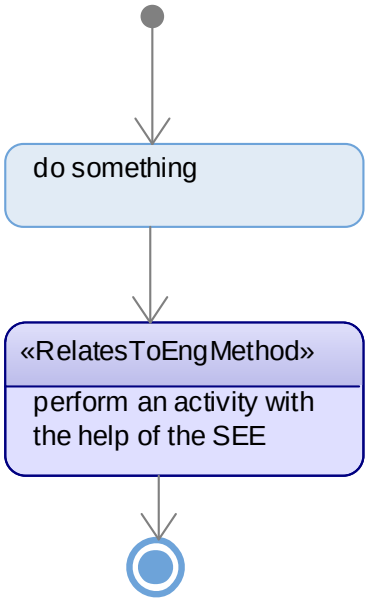

Figure 1-4 Activity Diagram – Description of Process Activity

A process activity usually comprises several activities performed by one or more project members. The sequences in the shown activity diagrams is according to our internal processes. Often they are simplified and are not necessarily identical or they depict only one possible sequence.

perform an activity with the help of the SEE

Within the sequence of activities, there are some activities, which are performed in the SEE especially focusing on the interoperability of the SEE integrated tools. These activities relates to an Engineering Method. The Engineering Method will describe the activities performed using the SEE in detail.

An Engineering Method describes how an activity can be realised using guidelines and tools which interoperate with each other. It can be applied to one or more activities. The relation of the Engineering Methods and the Process Activities is described in a use case diagram, see Figure 1-4.

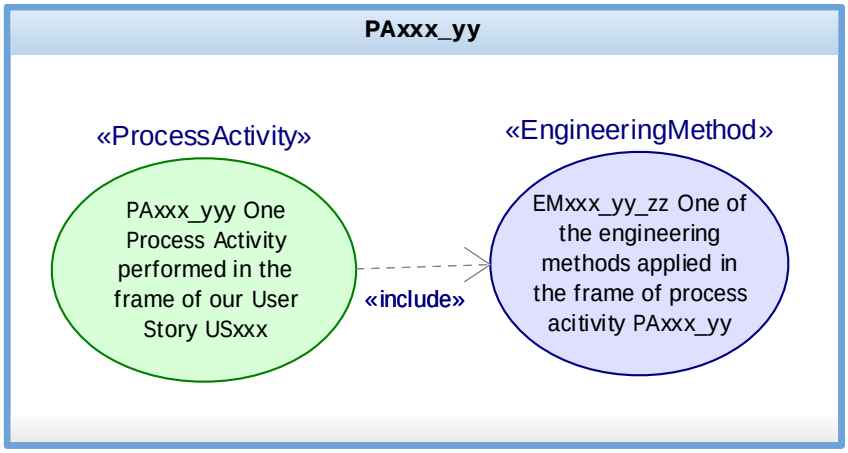

Figure 1-5 Use Case Diagram – Overview of Engineering Methods

Each Engineering Method is described by its purpose, pre-conditions, post-conditions and a set of engineering activities. One or more sequence diagrams (see Figure 1-5) depict possible scenarios of the Engineering Method using events sent between actors (e.g. SystemsEngineer) and generic tool functions (e.g. ReqManagement) as well as operations implemented by generic tool functions. For each sequence diagram, the operations are described in more detail. The description contains a reference to the related engineering activity. Events including the artefacts exchanged between the generic tool functions will be addressed in the next iteration of this document in order to provide a more precise definition for the Interoperability Specification (IOS).

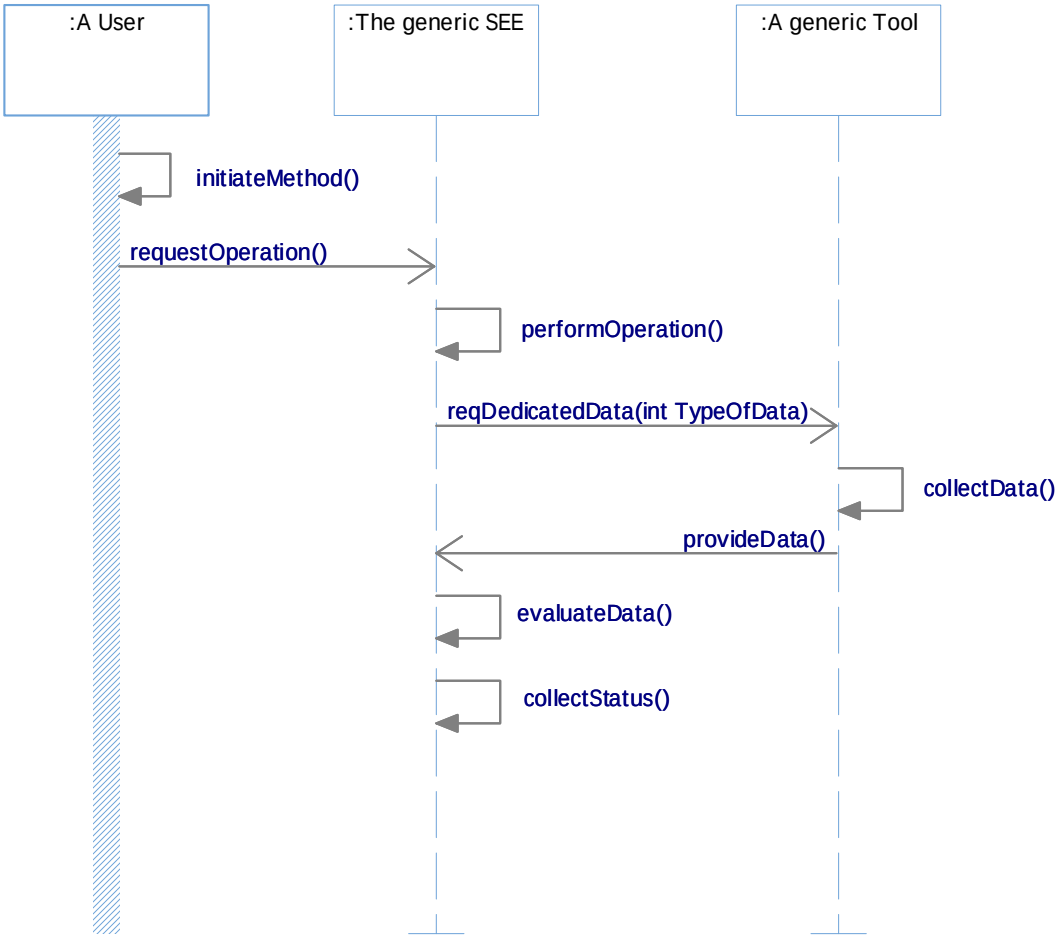

Figure 1-6 Sequence Diagram – Description of Engineering Methods

In order to support partners from the R&T work packages in sub-project SP6, several annexes have been attached to this document.

Annex I summarizes the identified Engineering Methods in tabular form using the template provided for the IOS Needs Capturing Process, see Figure 1-6. The upper part of the table provides general information on the Engineering Method such as the purpose, the pre- and post-conditions to be fulfilled as well as the flow of engineering activities to be performed. The lower part of the table describes the artefacts that need to be exchanged between tools when the engineering activities are performed. The description of the artefacts may contain information on the type or relevant attributes of the artefact and additional interoperability constraints.

Engineering Method:					
Purpose:					
Comments:					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Notes:		Notes:		Notes:	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:		Name:	-	Name:	
Generic Type:		Type:	-	Generic Type:	
Required Properties:		Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	

Figure 1-7 Engineering Method Template

Annex III comprises the requirements that shall be fulfilled by the CRYSTAL SEE from the application domain's perspective. The technology providers are supposed to analyze these requirements, refine and allocate them to the Reference Technology Platform (RTP), Interoperability Specification (IOS) and impacted technology bricks accordingly.

For each Engineering Method a requirements diagram is provided, see Figure 1-7. The requirements are related with the Engineering Method using a <<trace>> dependency. Subsequently, each requirement is described by a unique ID, its requirements text and additional information.

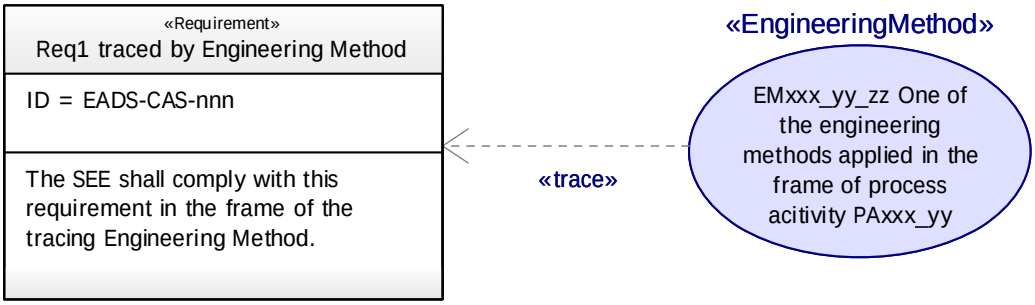

Figure 1-8 Requirements Diagram

Note: Exported with Rational Publishing Engine from the Rhapsody SysML Model, 2014-01-29

1.5 Open Points

EM - Perform Report Generation	The Engineering Method "Perform Report Generation" will be elaborated in the next version of the deliverable.
PA - Perform requirements engineering	The definition of ontologies, boilerplates, patterns, and metrics will be elaborated in the next version of the deliverable

2 Use Case Description

The use case focuses on situational awareness systems which support helicopter pilots to identify, process and comprehend mission or flight critical information.

2.1 Overview

One of today's main challenges in the aerospace domain is to enhance the pilot's situational awareness. This means, that the pilot shall be able to identify, process and comprehend mission/flight critical elements of information under all circumstances e.g.:

- Take-off
- Landing
- Low level flight
- Planning of cargo load
- Mission data preparation

Possible root causes for loosing situational awareness can be for example:

- Confusion caused by multiple simultaneous visual and/or acoustic alerts
- Missing information about potential hazards
- Failure to meet planned objectives
- Ambiguous information

The risk of losing situational awareness might have catastrophic consequences with regards to mission or even human life.

The key success factor will be to support the pilot with information in a simple, precise way, clearly understandable, and focused on the relevant information. Therefore it is very important to adapt the information level depending on the current mission/flight scenario.

The Mission Support Equipment (MSE) shall therefore support different mission/flight scenarios. This shall be achieved by several functions to support the pilot on ground and/or in flight, providing adequate visual or acoustic information using databases, sensor data and/or a data fusion of both sources for a certain phase of flight/mission.

2.2 Complexity and System Features

During studies on airplane and helicopter mission/flight scenarios and workshops with pilots following functions have been identified, having potential positive impact on situational awareness:

- MAP display
- Tactical symbology
- On-ground mission preparation (map data, overlays)
- Real-time update of mission relevant information (e.g. weather or traffic)
- Obstacle detection and warning (sensor data and database)
- Landing aid based on sensor data and/or database
- Sensor data and database supported vision enhancement under bad environment condition (e.g. rain, snow, fog, dust, night)

2.3 Landing Symbology Feature

For demonstration and evaluation purposes the Landing Symbology feature will be taken in this use case. The Landing Symbology supports helicopter pilots during the final landing approach in degraded visual environments which can be caused by e.g. rain, fog, sand, dust and snow (see Figure 2-1). Many accidents can be directly attributed to such degraded visual environments where pilots often loose spatial and environmental orientation.

Figure 2-1 Degraded Visual Environments

The Landing Symbology feature allows to mark the landing point on ground using a head-tracked helmet mounted display. During the final landing approach it enhances the spatial awareness of flying crews by displaying 3D conformal visual cues on a helmet-mounted display. In addition it employs a surface grid conformal to the measured terrain for the landing area.

The Landing Symbology feature provides the following functionality:

- display 3D conformal visual cues on a helmet mounted display visualizing the helicopter attitude and position relative to the intended landing point
- determine and visualize the condition of the anticipated landing zone with respect to roughness and slope based on real time 3D data
- display obstacles on a helmet mounted display relevant for the start and landing phase. The obstacles are taken from the real-time obstacle fusion, thus considering obstacles from the obstacle data bases and from real-time sensor obstacle classification.

Figure 2-2 shows an example of the landing symbology displayed during the final landing approach.

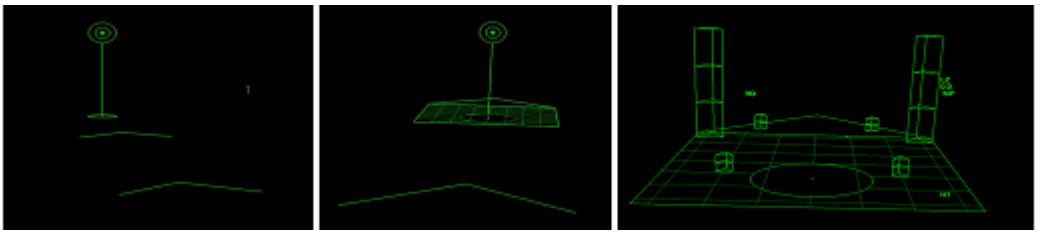

Figure 2-2 Display of the Landing Symbology

The Landing Symbology feature is part of a situational awareness system. It uses available information coming from databases and sensors. Figure 2-3 depicts the potential system architecture.

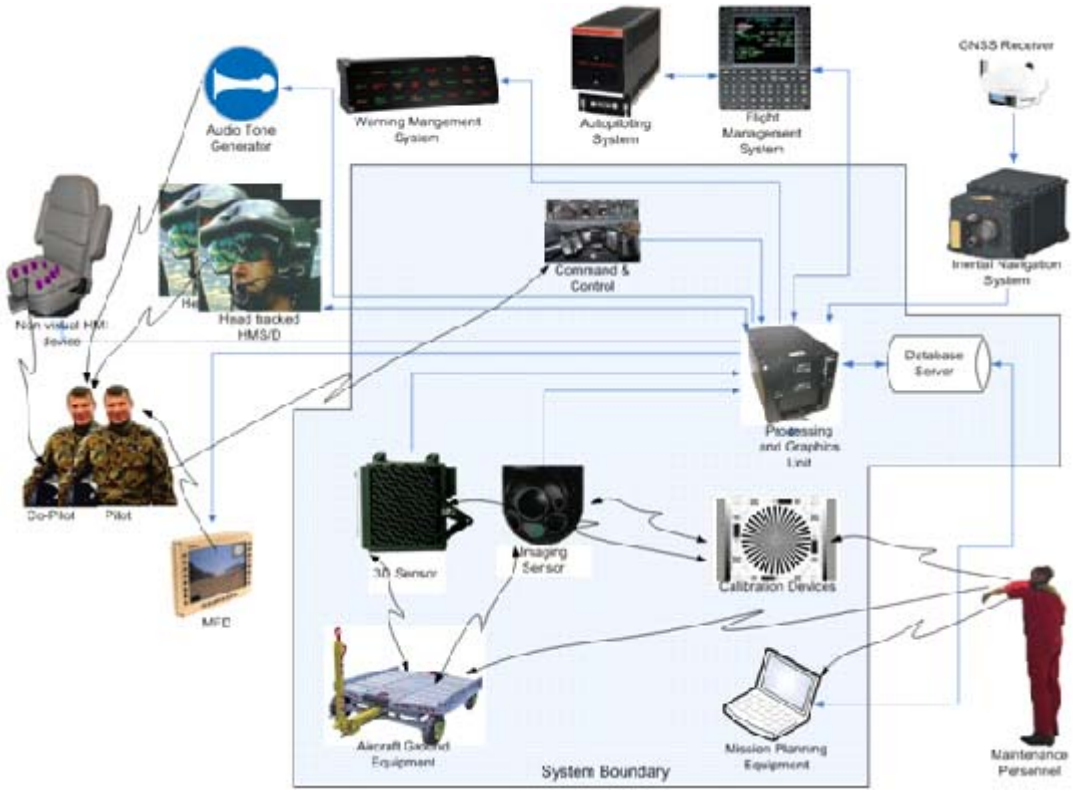

Figure 2-3 System Architecture of Situational Awareness Systems

3 Detailed Description of the Use Case Process

The Cassidian Use Case is mainly dealing with the following User Stories:

- Safety analyses
- Variability management
- Ontology-based requirements formalization and validation
- Process automation, guidance & monitoring
- Artefacts advanced traceability

The current technology baseline and the identified gaps that need to be addressed by CRYSTAL technical innovations and technology bricks are depicted in Annex I.

The following Use Case Diagram describes the Cassidian User Stories and their associations with the Crystal Partners.

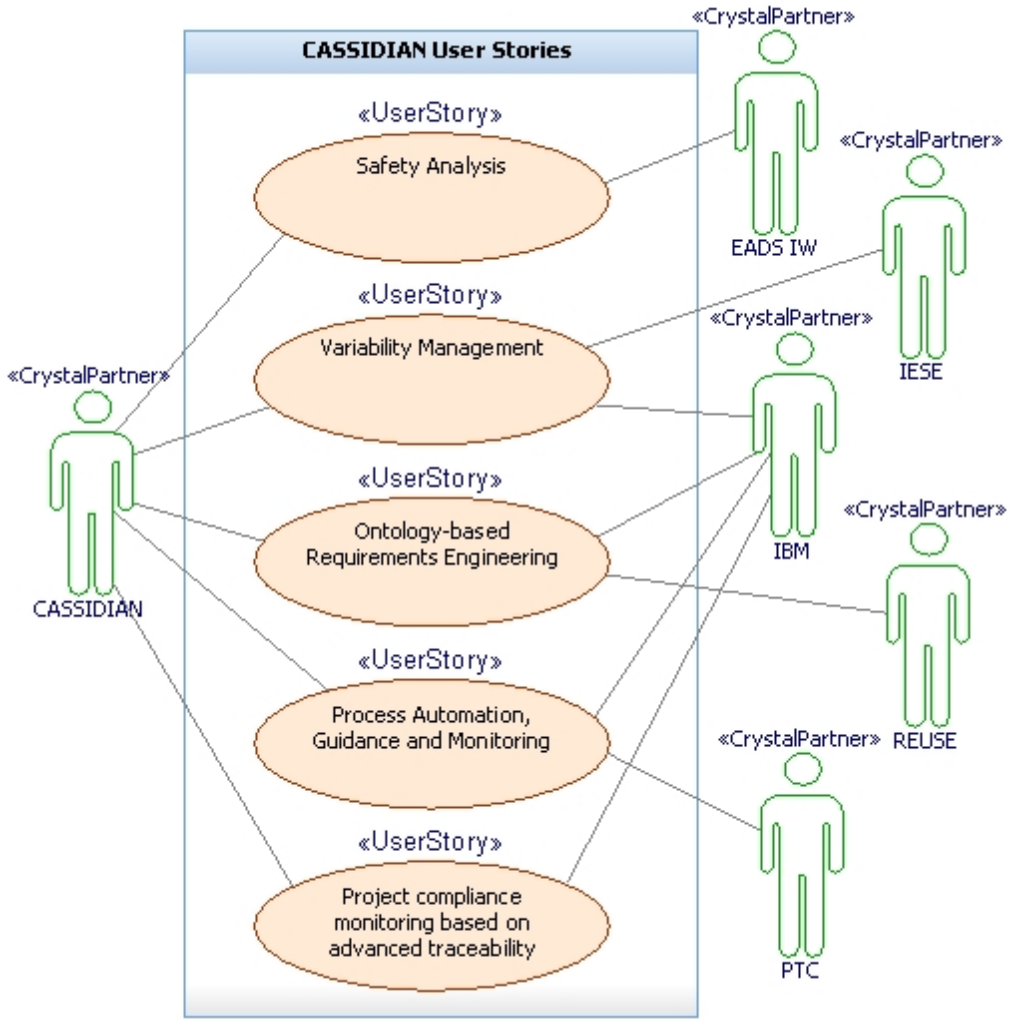

Figure 3-1 Cassidian User Stories

The Process Activities and the engineering methods of the particular User Story are identified and described in the following subchapters.

3.1 User Story US202 - Safety Analysis

Define a modelling approach to support aerospace safety analyses

- Define functions of system under design
- Define failure cases for the system functions
- Define criticality of functions
- Define candidate architecture to implement system functions with failure rates for candidate system components
- Analyse quantitative safety aspects
- Refine / change architecture if safety objectives have not been met.

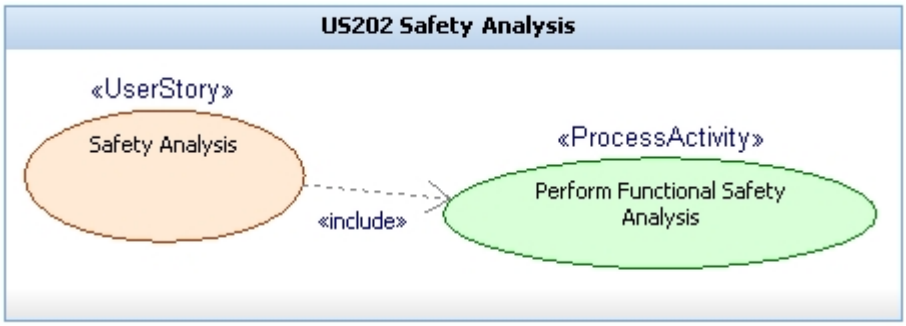

Figure 3-2 UCD_US202_SafetyAnalysis

The User Story Safety Analysis is including the Process Activities listed below. Note, that a Process Activity is detailed within another User Story if no <<include>> dependency is established from the User Story.

- Perform Functional Safety Analysis

3.1.1 PA202_01 - Perform Functional Safety Analysis

ID: PA202_01

Related User Story: Safety Analysis

The process activity "Perform Functional Safety Analysis" describes the systems engineering activities in conjunction with the safety engineering activities. Safety considerations are an integral part of the SE activities, which results in a high level of interrelation of safety activities with the functional analysis and architectural design activity.
Such safety analysis is conducted on the "white box" logical architecture or the physical architecture.

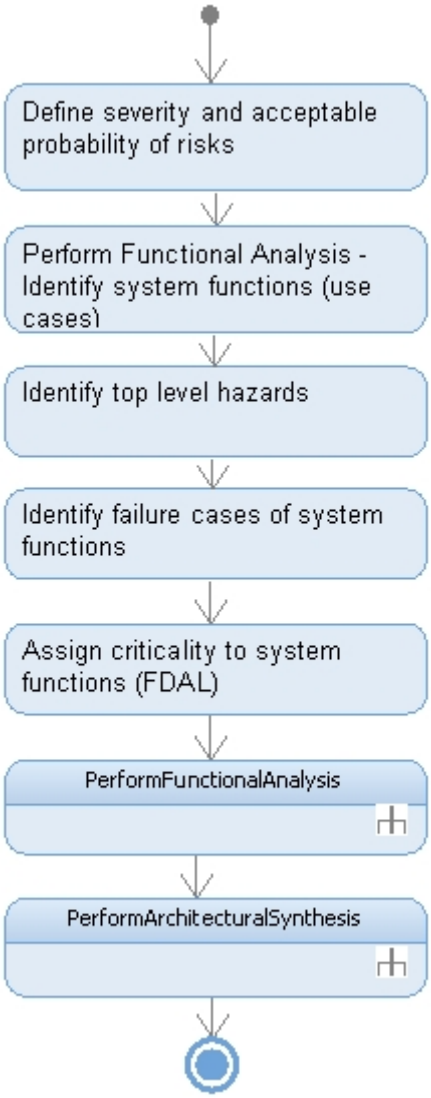

Figure 3-3 AD_PA202_01

Assign criticality to system functions (FDAL): The maximum severity of the identified failure causes determines the FDAL of the system use case. Assign it to the tagged value "FDAL" of the use case.

Define severity and acceptable probability of risks: Setup a cross reference list of severity levels and their associated risk probability as well as the applicable FDAL level. For avionic systems the ARP4754A might be applicable.

Identify failure cases of system functions: For each system use case the severity of failure for the individual failure cases is stored in the related tagged value "FHA*". Not applicable failure cases (of the list of default "FHA*" tags) get the value "no safety effect" assigned.

Identify top level hazards: Identify the top level hazards the system might cause.

Perform Functional Analysis - Identify system functions (use cases): System use cases are derived from the stakeholder requirements and/or ConOps.

PerformArchitecturalSynthesis: This action is a sub activity flow of the main activity diagram "AD_PA202_01" and is described by the Activity Diagram below.

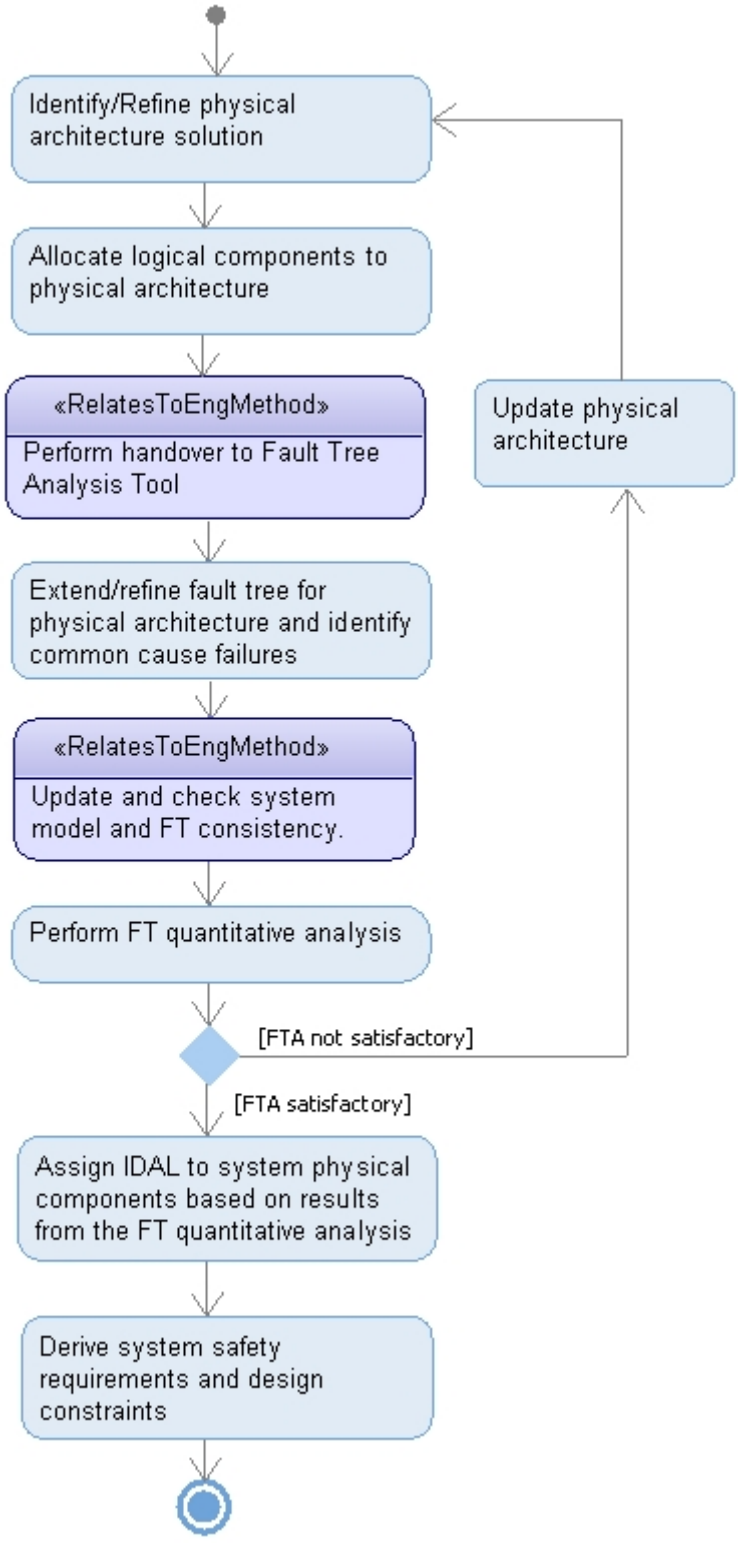

Figure 3-4 PerformArchitecturalSynthesis

Allocate logical components to physical architecture: Logical components (blocks) are implemented by physical components. This is expressed in accordance with the IBM Harmony SE via an <<implements>> dependency.

Assign IDAL to system physical components based on results from the FT quantitative analysis: The Item Development Assurance Level (IDAL) is defined in the ARP4754A.The IDAL is stored in the system model (SysML) tagged value "IDAL" of physical blocks.

Derive system safety requirements and design constraints: Derived system requirements are defined in the requirements management tool and traced to the respective system model element.

Extend/refine fault tree for physical architecture and identify common cause failures: Define fault tree for physical (including logical) architecture and identify common cause failures.

Identify/Refine physical architecture solution: Define system Physical Architecture (in BDD) in accordance with IBM Rational Harmony SE Best Practice.

Perform FT quantitative analysis: Perform FTA and elicit a consistent set of reliability figures for the defined functions and there reliability figures.

Perform handover to Fault Tree Analysis Tool (RelatesToEngMethod): Transfer relevant system model data to FTA tool for further analysis.

Relates to Engineering Method: Set_Reliability_and_Handover_Physical

Update and check system model and FT consistency. (RelatesToEngMethod): Check and report on consistency between system model (SysML) and the Fault Tree definition. Goal is, to have a consistent set of functions, failure cases, severity levels and reliability figures defined in the two models.

Relates to Engineering Method: Check_Model_and_FT_Consistency_Physical

Update physical architecture: Update the functional architecture with the knowledge gained.

PerformFunctionalAnalysis: This action is a sub activity flow of the main activity diagram "AD_PA202_01" and is described by the Activity Diagram below.

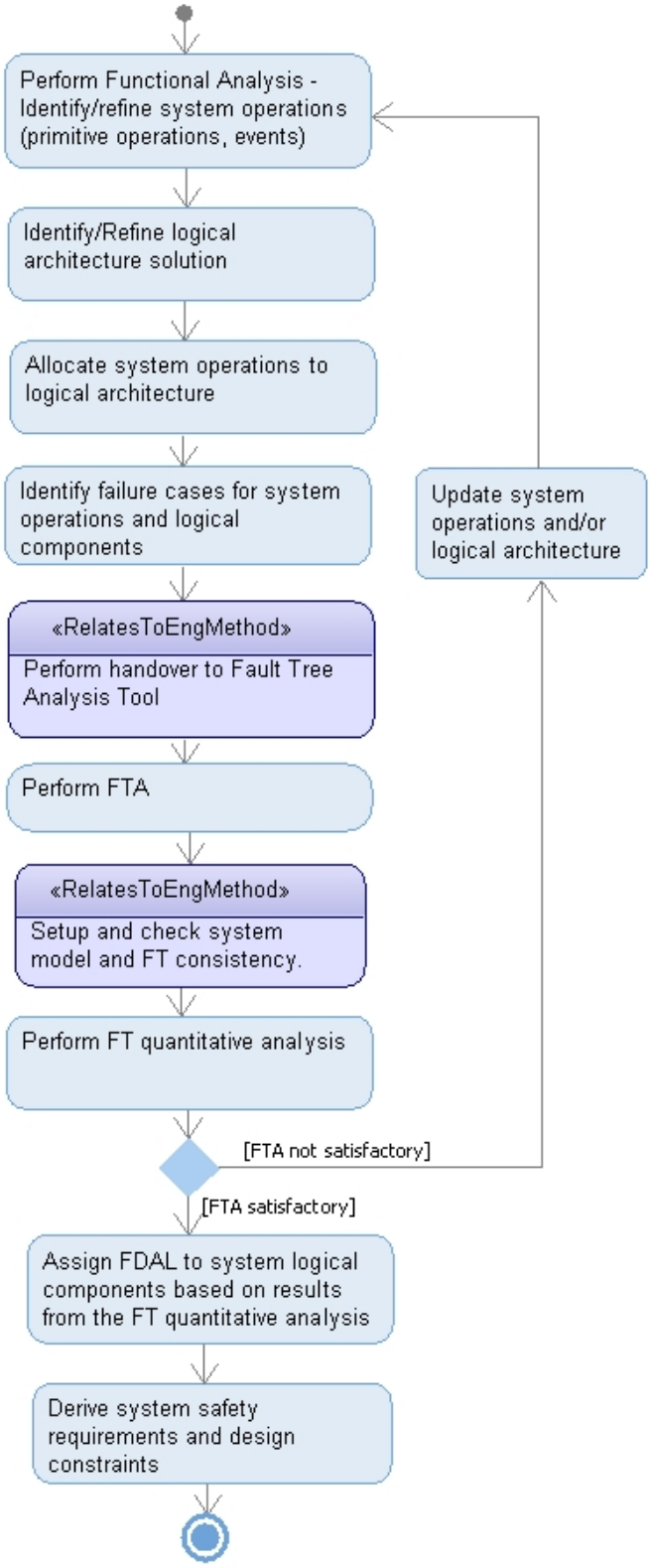

Figure 3-5 PerformFunctionalAnalysis

Allocate system operations to logical architecture: Perform Logical Analysis (functional, white box) in accordance with IBM Rational Harmony SE Best Practice.

Assign FDAL to system logical components based on results from the FT quantitative analysis: The Function Development Assurance Level (FDAL) is defined in the ARP4754A. The FDAL is stored in the system model (SysML) tagged value "FDAL" of logical blocks and primitive operations.

Derive system safety requirements and design constraints: Derived system requirements are defined in the requirements management tool and traced to the respective system model element.

Identify failure cases for system operations and logical components: Assign the severity level to each block on primitive operation failure case. The failure cases are stored as tagged value with the prefix "FHA_". By default they are:

- FHA_Erroneous
- FHA_Inadvertant
- FHA_Late
- FHA_Loss
- FHA_PartialLoss

Identify/Refine logical architecture solution: Define system Logical Architecture (in BDD) in accordance with IBM Rational Harmony SE Best Practice.

Perform FT quantitative analysis: Perform FTA and elicit a consistent set of reliability figures for the defined functions and there reliability figures.

Perform FTA: Define fault tree for logical architecture and identify common cause failures.

Perform Functional Analysis - Identify/refine system operations (primitive operations, events): Perform system Functional Analysis (black box) in accordance with IBM Rational Harmony SE Best Practice.

Perform handover to Fault Tree Analysis Tool (RelatesToEngMethod): Transfer relevant system model data to FTA tool for further analysis.

Relates to Engineering Method: Perform_FTA_Handover

Setup and check system model and FT consistency. (RelatesToEngMethod): Check and report on consistency between system model (SysML) and the Fault Tree definition. Goal is, to have a consistent set of functions, failure cases, severity levels and reliability figures defined in the two models.

Relates to Engineering Method: Check_Model_and_FT_Consistency_Logical

Update system operations and/or logical architecture: Update the functional architecture with the knowledge gained.

3.1.1.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

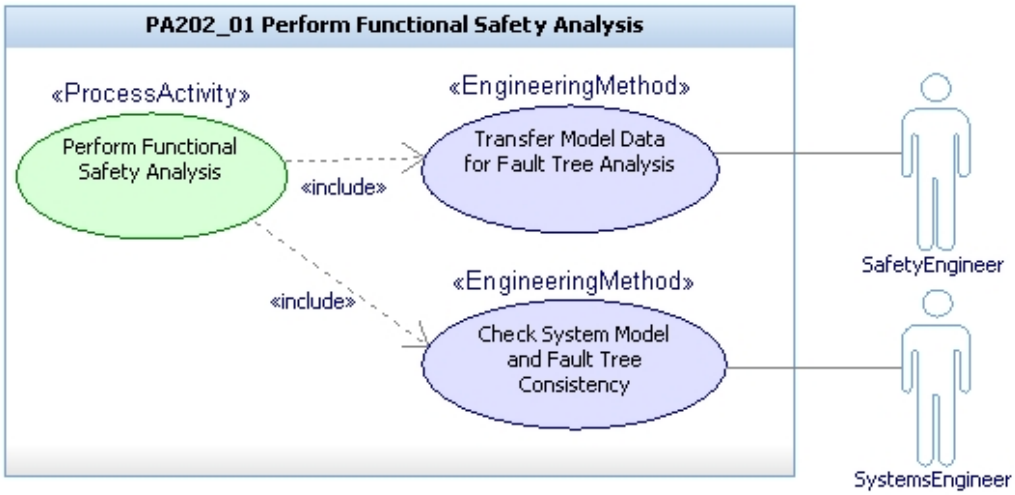

Figure 3-6 UCD_PerformFunctionalSafetyAnalysis

3.1.1.2 EM202_01_01 - Transfer Model Data for Fault Tree Analysis

ID: EM202_01_01

The system model structure has to be according to the IBM Harmony SE. It is assumed that within the fault tree tool (like FT+) basic elements need to be created, representing the elements from the system model. A trace dependency has to be established between the fault tree basic elements to the represented system model elements. Dedicated tags are used for the system model elements to receive the results from the fault tree analysis.

EM Purpose: Logical blocks with their primitive operations and/or the physical blocks from the system model are made available in the fault tree tool in order to create representations for the fault tree analysis.

EM Pre-Condition: Logical architecture with primitive operations and/or physical architecture are available in the system model.

EM Post-Condition: Model elements transformed into a set of artefacts (basic elements) as input for fault tree analysis.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Extract System Use Cases from model
02. Select Use Case and type for FTA
03. Extract physical architecture data (physical FTA only)
04. Extract logical architecture data
05. Generate basic elements library
06. Establish traces to originating elements

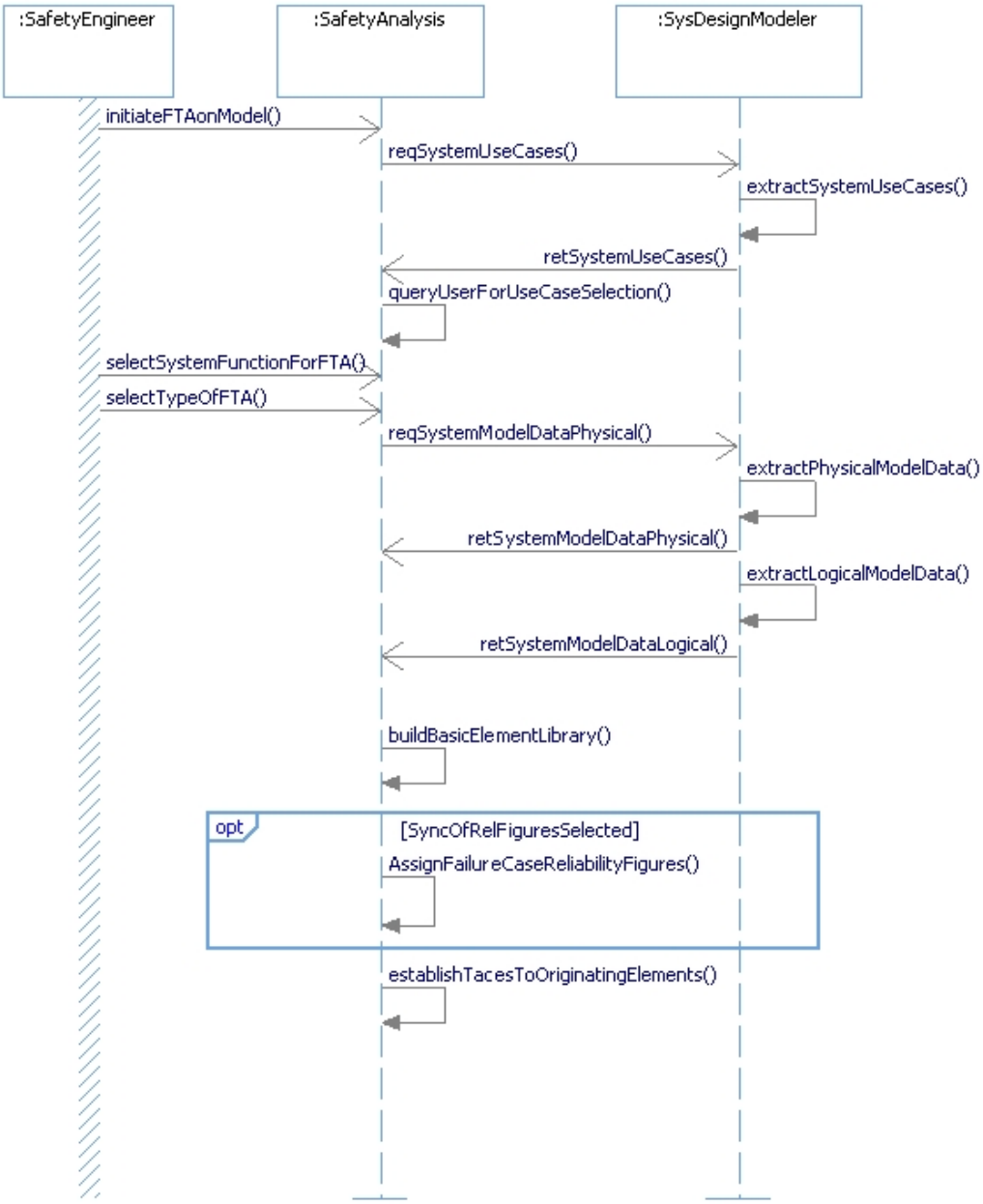

Figure 3-7 SD_EM202_01_01_PhysicalFTA

AssignFailureCaseReliabilityFigures: The severity, stored as "FHA*" tagged value, shall be assigned with its corresponding failure probability to the respective failure cases (e.g. for validation purpose) of the use cases, blocks and functions.

Note: The mapping of severity to corresponding FDAL and failure probabilities has to be stored within the systems model.

buildBasicElementLibrary: The received elements are translated into basic elements. Each basic element gets a generic failure case associated per defined "FHA*" tagged value associated with the model element. Those "FHA*" tagged values are ignored which have "no safety effect" assigned. The basic elements and failure cases are stored in a basic element library for fault tree definition.

Relates to Engineering Activity: 05. Generate basic elements library

establishTacesToOriginatingElements: For each basic element a trace dependency back to the origin system model element is established. This allows

re-synchronization of the models in case the system model is changed,

to perform completeness and consistency checks and

to feed back the results of the fault tree analysis into the system model.

Relates to Engineering Activity: 06. Establish traces to originating elements

extractLogicalModelData: The model elements of interest to be extracted are:

- Use Case contained in FunctionalAnalysisPkg (or sub-packages) including its tagged values "FHA*"

- Blocks contained in the LogicalAnalysisPkg (or sub-packages) including its tagged values "FHA*"

- Primitive Operations contained by each block including its tagged values "FHA*"

+ all in case of complete system selected by user

+ filtered to the use case relevant functions (use case traces to dedicated white box activity diagram, containing actions named as the relevant operations).

Blocks without operations are to be ignored.

For each model element the unique identifier has to be transferred as well (e.g. GUID).

Relates to Engineering Activity: 04. Extract logical architecture data

extractPhysicalModelData: Blocks (or derived model elements) belonging to the physical architecture are located in the PhysicalAnalysisPkg. The extraction gathers the system and all of its system elements down to the lowest level of hierarchy. For each block extract the associated reliability figures for the different failure cases, stored as tagged value "FHA*".

For each model element the unique identifier has to be transferred as well (e.g. GUID)

Relates to Engineering Activity: 03. Extract physical architecture data (physical FTA only)

extractSystemUseCases: The modelling tool shall query the system model for use cases, located in the package "FunctionalAnalysisPkg" (or sub-packages).

Relates to Engineering Activity: 01.Extract System Use Cases from model

queryUserForUseCaseSelection: The safety analysis tool shall present the available use cases to the user for selection of the relevant one. Note: A use case is seen equivalent to a system function.

Relates to Engineering Activity: 02. Select Use Case and type for FTA

Figure 3-8 SD_EM202_01_01_LogicalFTA

AssignFailureCaseReliabilityFigures: The severity, stored as "FHA*" tagged value, shall be assigned with its corresponding failure probability to the respective failure cases (e.g. for validation purpose) of the use cases, blocks and functions.

Note: The mapping of severity to corresponding FDAL and failure probabilities has to be stored within the systems model.

buildBasicElementLibrary: The received elements are translated into basic elements. Each basic element gets a generic failure case associated per defined "FHA*" tagged value associated with the model element. Those "FHA*" tagged values are ignored which have "no safety effect" assigned. The basic elements and failure cases are stored in a basic element library for fault tree definition.

Relates to Engineering Activity: 05. Generate basic elements library

establishTacesToOriginatingElements: For each basic element a trace dependency back to the origin system model element is established. This allows

re-synchronization of the models in case the system model is changed,

to perform completeness and consistency checks and

to feed back the results of the fault tree analysis into the system model.

Relates to Engineering Activity: 06. Establish traces to originating elements

extractLogicalModelData: The model elements of interest to be extracted are:

- Use Case contained in FunctionalAnalysisPkg (or sub-packages) including its tagged values "FHA*"

- Blocks contained in the LogicalAnalysisPkg (or sub-packages) including its tagged values "FHA*"

- Primitive Operations contained by each block including its tagged values "FHA*"

+ all in case of complete system selected by user

+ filtered to the use case relevant functions (use case traces to dedicated white box activity diagram, containing actions named as the relevant operations).

Blocks without operations are to be ignored.

For each model element the unique identifier has to be transferred as well (e.g. GUID).

Relates to Engineering Activity: 04. Extract logical architecture data

extractSystemUseCases: The modelling tool shall query the system model for use cases, located in the package "FunctionalAnalysisPkg" (or sub-packages).

Relates to Engineering Activity: 01.Extract System Use Cases from model

queryUserForUseCaseSelection: The safety analysis tool shall present the available use cases to the user for selection of the relevant one. Note: A use case is seen equivalent to a system function.

Relates to Engineering Activity: 02. Select Use Case and type for FTA

3.1.1.3 EM202_01_02 - Check System Model and Fault Tree Consistency

ID: EM202_01_02

This engineering method allows to check and report on the consistency between the system model (SysML) and the corresponding fault tree model.

EM Purpose: Check that the functions, related failure cases and severity classification is consistent to the fault tree elements and their related reliability figures.

EM Pre-Condition: FTA performed

EM Post-Condition: Traceability between model and fault tree elements established

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Set System Model Scope
02. Retrieve FTA Data
03. Correlate System Model Elements With FTA Elements
04. Compare Reliability Figures
05. Generate Safety Coherence Report

Figure 3-9 SD_EM202_01_02_FT Consistency Check

compareReliabilityFigures: The tool shall perform a smart compare of the reliability figures. The system model element (specification character) reliability figures may be more stringent than the figures of the FTA or rounded, still providing a positive match.

Relates to Engineering Activity: 04. Compare Reliability Figures

correlateSystemModelElementsWithFTAElements: Correlate the received FT element data with the use cases, blocks and primitive functions of the system model.

Identify system model elements (blocks, primitive operations, use cases) as well as FT elements not covered by the corresponding model.

A filter shall be available to consider only basic (leaf) elements of the FT.

Relates to Engineering Activity: 03. Correlate System Model Elements With FTA Elements

generateSafetyCoherenceReport: Generate suitable reports (lists, documents and/or traceability matrix) to the systems engineer for further actions.

Relates to Engineering Activity: 05. Generate Safety Coherence Report

retrieveFTAData: Based on the selection criteria a list of FT element data is created, including the failure cases and the calculated reliability figures and traceability information.

Relates to Engineering Activity: 02. Retrieve FTA Data

setSystemModelScope: The user selects the model scope based on the active workspace/model.

The scope might be logical / physical architecture and the complete system or a specific system function, i.e. Use Case.

Relates to Engineering Activity: 01. Set System Model Scope

3.2 User Story US203 - Variability Management

Define a process for the identification, specification and tracing of variability points to development artefacts

- Model common and variable feature dependencies in a feature model.
- Create a variability model covering external and internal variability.
- Link the variability model with Requirements.
- Link the variability model with System Architecture Model
- Develop variability model configurations to provide consistent visibility to external customer features and to external and internal variability.
- Create and manage components and systems model library and combine these models together.
- Set up specific modelling guidelines and define associated modelling standard, if necessary.

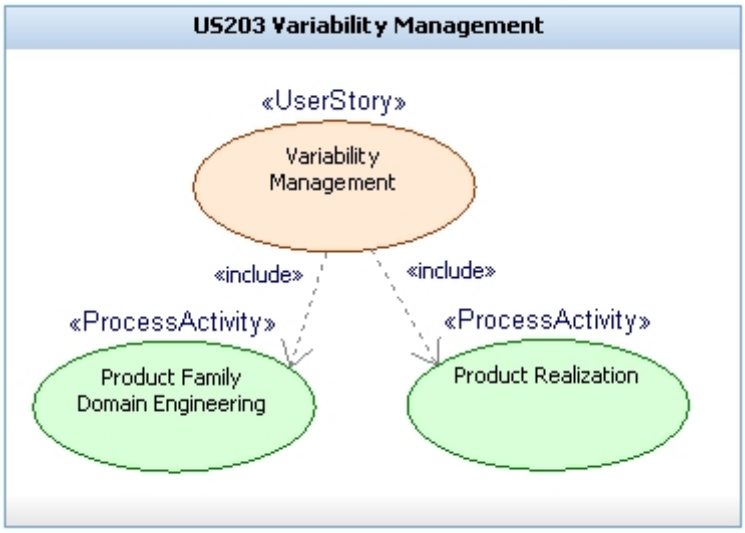

Figure 3-10 UCD_US203_VariabilityMgmt

The User Story Variability Management is including the Process Activities listed below. Note, that a Process Activity is detailed within another User Story if no <<include>> dependency is established from the User Story.

- Product Family Domain Engineering
- Product Realization

3.2.1 PA203_01 - Product Family Domain Engineering

ID: PA203_01

Related User Story: Variability Management

The domain engineering develops all assets which shall be reused in scope of the product family. In the CRYSTAL project we concentrate only on the indicated engineering methods, i.e. "Develop domain system requirements" and "Develop reference architecture".

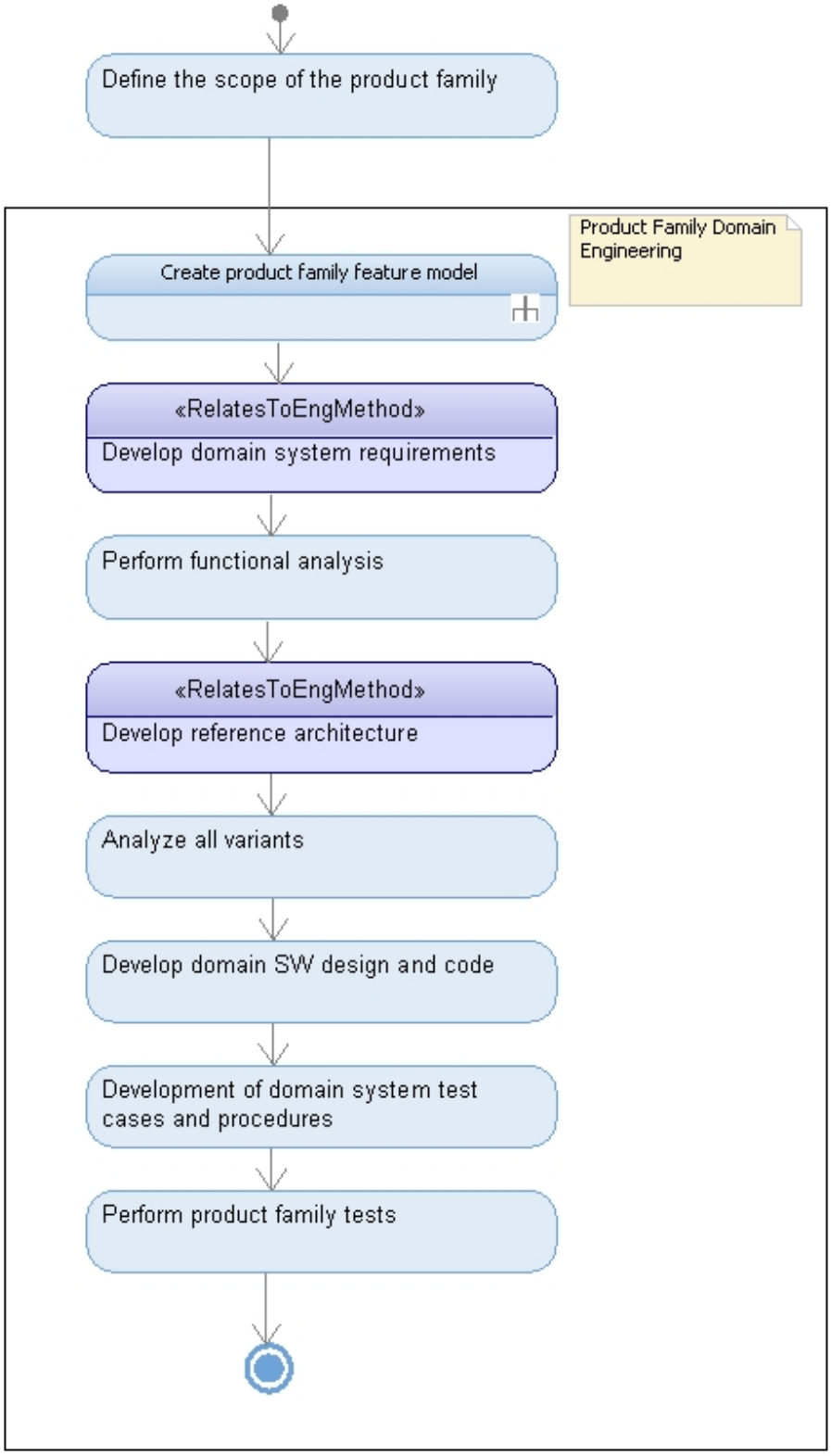

Figure 3-11 AD_PA203_01

Analyze all variants: Analysis of all possible variants. Indicate and check for model gaps or inconsistencies.

Define the scope of the product family: Based on business information features are identified and future product configurations are defined.

1.) Feature Identification:

Identify features which could provide value to at least one customer.

2.) Feature Classification

Features and options shall be classified according AVAILABILITY and COMMONALITY.

3.) Feature relevance assessment:

Assessing the relevance of a feature shall consider:

- value or importance of the feature for the customer
- costs to implement that feature
- readiness of the feature

Each feature shall be assessed and represented in the product roadmap.

4.) Scope definition:

Define if a feature is IN or OUT of scope. Indicate the "InScope" features as well as the "OutOfScope" features and provide the rationales for the decision. Keeping these rationales in visible to the entire project might avoid unintended scope discussions during the product life realization.

5.) Future product definition

Based on business information top level product configurations shall be established for future products. Product configurations shall describe a low end and a high end configuration or a low cost and a premium configuration.

A product configuration shall list the required features and required variants.

Develop domain SW design and code: HW/SW design and coding of SW assets is performed for all variants within the product family scope.

Develop domain system requirements (RelatesToEngMethod): The system requirements for the entire product family shall be developed in accordance to the product family scope.

Develop reference architecture (RelatesToEngMethod): In the system architecture development of a product family common and variable model elements shall be identified and linked to the feature model.

The physical analysis identifies external physical interfaces, elements of the physical architecture, properties of physical components (e.g. memory size), COTS suppliers.

Development of domain system test cases and procedures: Test cases and procedures are developed on System and SW/HW level for all defined product variants within the product family.

Perform functional analysis: The functional analysis serves for the identification of functions and sub-functions. The functional scope, the interactions with operators and external system, functions and sub-functions of the system shall be identified.

Perform product family tests: As much as possible testing credit shall be created during domain engineering instead of testing it for every product variant realization again and again.

This is mainly driven by the variations. If especially the HW platform is variable or main SW components are replacing each other more testing credit might get lost.

AD_PA203_01_01_Create product family feature model: This action is a sub activity flow of the main activity diagram "AD_PA203_01" and is described by the Activity Diagram below.

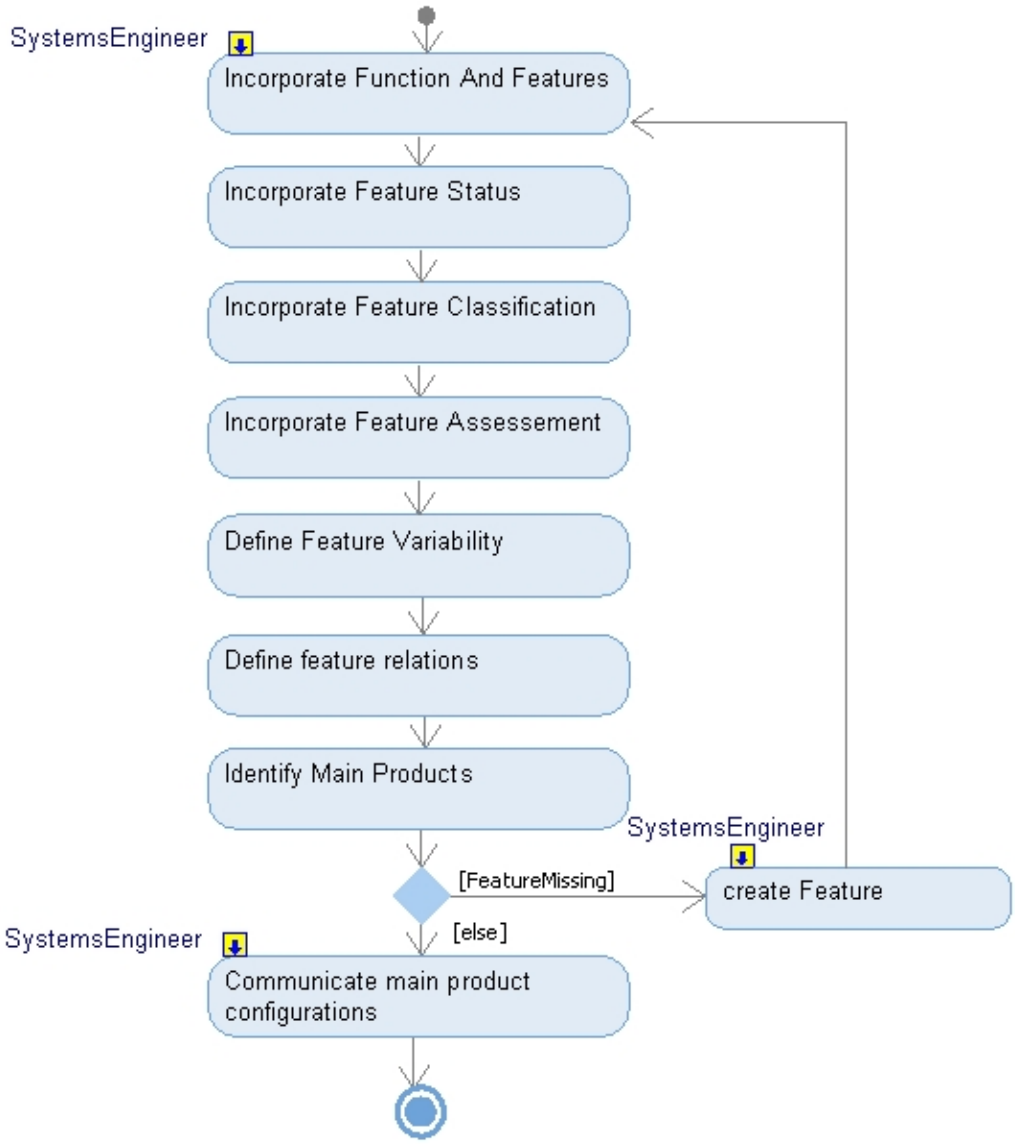

Figure 3-12 AD_PA203_01_01_Create product family feature model

Communicate main product configurations: The main product configuration shall be stored and communicated in accordance with the product roadmap.

create Feature: If a required feature is missing it shall be created.

Define feature relations: If a feature or option is required for a dedicated product other features might be affected. These relations shall be identified in the feature model.

1. Relations between variation points might exist.

2. Relations between options might exist.

3. Relations between an option and other variation points might exist.

These relations will be indicated to the user during the creation of a product configuration. Especially in case of constraints awareness needs to be provided to the SEEuser.

Define Feature Variability: The variability within the product family shall be indicated by variation points and related options. For each variation point the possible options of the variability shall be defined.

Each variation point shall be marked if it shall be part of the external view provided e.g. for customers negotiations.

Identify Main Products: The product bandwidth of the product family shall be expressed. E.g. the low and high end product configurations shall be identified. If already a more detailed visibility on the first products of the product roadmap is available these main product configurations shall be defined in the feature model indicating the required features.

Incorporate Feature Assessement: The feature assessment result shall be incorporated into the feature model for each feature and feature option.

Incorporate Feature Classification: The feature classification information defined in the scope definition activity shall be incorporated.

Incorporate Feature Status: The status of each feature identified during scoping process shall be incorporated.

- "new" as default if status is not defined
- "development" if is already under development
- "planned" if the development is already scheduled.
- "hypothetical" if it is a candidate in the future.

The status shall be maintained when it changes in the feature model.

Incorporate Function And Features: During scoping process a list of all functions and related features is created. The functions and features shall be incorporated into the feature model.

3.2.1.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

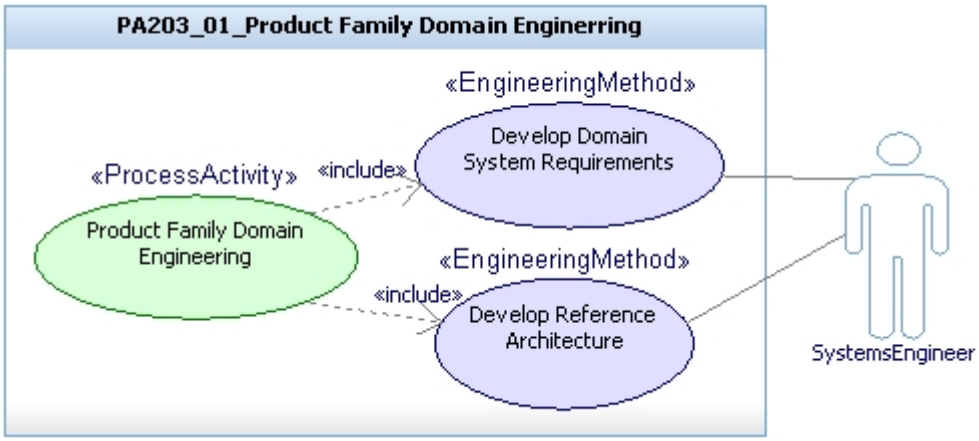

Figure 3-13 UCD_ProductFamilyDomainEngineering

3.2.1.2 EM203_01_01 - Develop Domain System Requirements

ID: EM203_01_01

The system requirements for the entire product family shall be developed considering the products to be achieved by the product family scope. The system requirements developed shall be linked to features and feature options.

Common system requirements which are equal to all product variants shall be linked to common features. Requirements which might be included or excluded between products shall be linked to a variable feature.

EM Purpose: The variability of the product family system requirements shall be managed consistently to variable functions, features.

EM Pre-Condition: System Requirements defined for a product family. Optional requirements are identified. Variable system requirements identified by variation points. Functions and features modelled in a feature model. variable functions and features identified.

EM Post-Condition: System Requirements are linked to features. System requirements variation points are defined, options are specified. Relations between features and variation points are described.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. create product family system requirements
02. define system requirements variation points
03. request functions and features of the product family
04. extract functions features of the product family
05. provide functions and features requirements data of the product family
06. display functions, features and allocated requirements
07. allocate system requirements to common features or feature options
08. link requirements variation points to features

09. check linkage between requirements and features

Figure 3-14 SD_EM203_01_01_DomainSystemRequirements

createProductFamilySystemRequirements: The product family system requirements for a feature shall be created. This includes optional or alternative system requirements. Optional or alternative requirements which are selectable for a product variant shall be linked to one feature option.

Each common system requirement shall be linked to a feature.

Relates to Engineering Activity: 01. create product family system requirements

defineSystemReqVariationPoints: Variability within system requirements e.g. thresholds which effects the system architecture needs to be resolved prior creation of the product variant. Therefore a [Variation Point] within the system requirements needs to be specified. Variation point options shall link to the system requirements.

Relates to Engineering Activity: 02. define system requirements variation points

displayAllocatedRequirements: Requirements which are already allocated to provided features or feature options shall be identified and displayed.

A list of available functions and features, showing the feature availability, feature commonality shall be displayed. Already allocated requirements per feature shall be displayed. Especially the allocation of feature options to system requirements shall be displayed to show which requirement is selected by which feature option.

Relates to Engineering Activity: 06. display functions, features and allocated requirements

linkRequirementsToFeatures: System requirements shall be linked either to common features or feature options. System requirements of a variable feature shall be allocated to feature options. If the allocated feature/option has any relation to another feature/option, the relation shall be indicated highlighting the effected type of relation e.g. requires, excludes.

The link information of system requirements to features shall be stored in the requirements database.

Relates to Engineering Activity: 07. allocate system requirements to common features or feature options

linkReqVariationPointsToFeatures: System requirement variation points shall be linked to the features or feature options of feature model.

Relates to Engineering Activity: 08. link requirements variation points to features

performRequirementsCheck: Tool supported model checks to identify inconsistencies shall be performed:

- if there are requirements without an linked feature or feature option.
- if requirements are allocated to deleted features or options (orphan).
- if requirements are allocated to changed features or options (suspect).

Relates to Engineering Activity: 09. check linkage between requirements and features

provideFunctionsAndFeatureList: The functions and feature information shall be provided:

- functions
- features
- feature availability (mandatory, optional, alternative)
- feature commonality (common or variable features)
- feature options
- relations between features and options
- feature model version

3.2.1.3 EM203_01_02 - Develop Reference Architecture

ID: EM203_01_02

In the system architecture development for product families variable artefacts shall be identified and linked to the feature model. The allocation of system model elements to features, variation points and options shall be established.

EM Purpose: The variability of the product family system model shall be managed consistently to variable functions and features.

EM Pre-Condition: The system architecture is developed. Functions and features variability is defined.

EM Post-Condition: The system model artefacts are linked to functions and features of the feature model. Resolution rules have been created in the system architecture model.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. define logical system view
02. validate logical system view
03. merge system states
04. create reference architecture
05. identify variable system model elements
06. request product family functions and features
07. extract product family functions and features
08. provide product family functions and features
09. link system model elements to features or feature options
10. complete resolution conditions of system model variation points
11. perform model check

Figure 3-15 SD_ReferenceArchitectureSynthesis

completeResolutionConditions: Variability has different effects to the system model elements. E.g.:

- existence: an object e.g. a port or a logical block exists or not

- value assignment: a variable value is assigned differently e.g. a threshold

- substitution: one object substitutes another one

- configuration unit: a different config file is required

The resolution conditions of the variable elements shall be completely defined in the system model. The rules and constraints how model elements are effected by feature model decisions shall be provided. E.g. A selected feature requires a decision to select between alternative implementation operations. Alternative operations might require for example different states and additional interfaces.

Relates to Engineering Activity: 10. complete resolution conditions of system model variation points

createReferenceArchitecture: The structure and the texture of the products in the product family shall be determined in the reference architecture.

The structure determines the static and dynamic decomposition that is valid for all products of the product family. It also indicates the common and variable elements and interfaces of the reference architecture.

The texture is the collection of common rules, guiding the design and realization of building blocks and the integration of those building blocks to from a product.

Relates to Engineering Activity: 04. create reference architecture

defineLogicalView: The functional decomposition into logical blocks using Block Definition Diagrams and Internal Block Diagrams shall be performed.

The logical activity flow shall be defined and depicts the activities allocated to logical blocks an the interactions between.

The sequential behaviour of logical elements shall be defined. The interactions between actors and internal logical blocks shall be represented as a white box scenario in a sequence diagram.

Logical external and internal interfaces shall be identified. A useful modelling method is the usage of Internal Block Diagrams.

The states and transitions of each logical blocks shall be defined using state charts.

Relates to Engineering Activity: 01. define logical system view

extractFeaturesAndOptions: All functions, the related features and options shall be extracted from the feature model. Also the relations between features and options shall be exported.

Relates to Engineering Activity: 07. extract product family functions and features

identifyVariableElements: Variability within the System Model shall be specified by [Variation Points].

Variable system model elements e.g. blocks, operations, interfaces, states, bitmaps shall be linked to the variation point options.

Relates to Engineering Activity: 05. identify variable system model elements

linkSystemModelElementsToFeatures: Link common and variable model elements to the related features.

Relates to Engineering Activity: 09. link system model elements to features or feature options

mergeStates: The separate modes and transitions of the logical blocks shall be merged to define the states of the entire system consistently.

Relates to Engineering Activity: 03. merge system states

performModelCheck: Basic checks shall be performed:

- Indicate model elements without feature or feature option allocation.

- Indicate if model elements are allocated to deleted features or options (orphan).

- Indicate if model elements are allocated to changed features or options (suspect).

Relates to Engineering Activity: 11. perform model check

provideFeatureAndOptions: The feature information shall be provided.

Relates to Engineering Activity: 08. provide product family functions and features

validateLogicalView: The logical view is validated by checking visually the state machine against the sequence diagram.

Relates to Engineering Activity: 02. validate logical system view

3.2.2 PA203_02 - Product Realization

ID: PA203_02

Related User Story: Variability Management

The product realization creates a product variant reusing the assets developed during domain engineering.

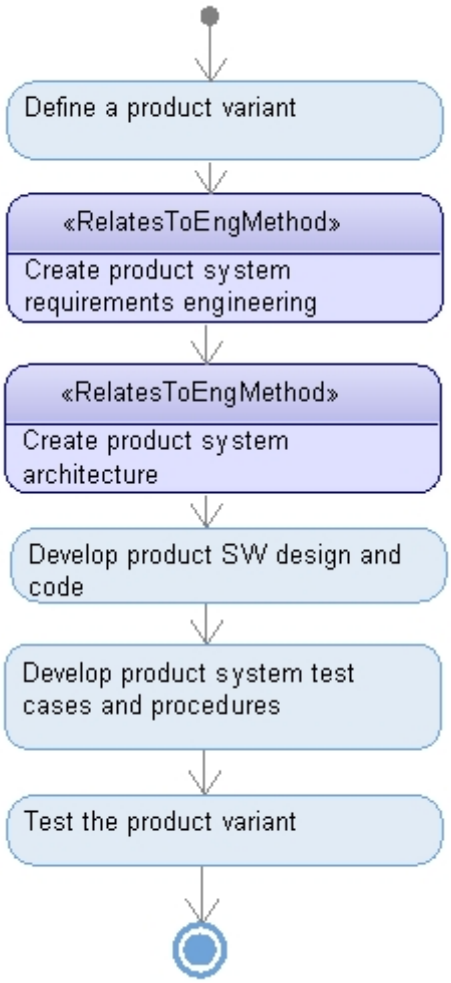

Figure 3-16 AD_PA203_02

Create product system architecture (RelatesToEngMethod): The product system architecture for a product variant shall be created according to the features and options described in the [product variant description].

In the product system architecture there might be internal [variation point]s which shall be resolved. The product variant description shall be enhanced by the selected options.

Create product system requirements engineering (RelatesToEngMethod): Based on the features in the [product variant description] the relevant system requirements for a product variant shall be identified and exported from the product family system requirements database into the product requirements documentation.

Variation points in the selected system requirements shall be resolved for the product variant. The common and variable system requirements shall be identified for the product variant.

Define a product variant: The variability within the product family feature model shall be resolved to create a product variant.

The list with all common features of the product shall be highlighted.

Optional features shall be highlighted and shall be resolved.

The feature resolution shall consider the feature relevance information and propose the most relevant features first.

A product variant description shall list all selected product features and feature options. The product variant description shall be reviewed for completeness and correctness. The CM identification and version of the product family feature model shall be stored in the product variant description to keep tracing between feature model version and product variant description.

Develop product SW design and code: SW design and code developed during the domain engineering are reused. Based on the selected features the SW Design and source code parts are identified and provided as a copy to the product realization.

Develop product system test cases and procedures: Based on the selected features in the [product variant description] the system test cases and procedures are identified and provided as a copy to the product realization.

Test the product variant: During product family domain engineering test results shall be created. Based on the selected features in the [product variant description] the relevant system test results are identified and provided as a copy to the product realization. It shall be analyzed which tests have to be repeated in the context of the product variant.

3.2.2.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

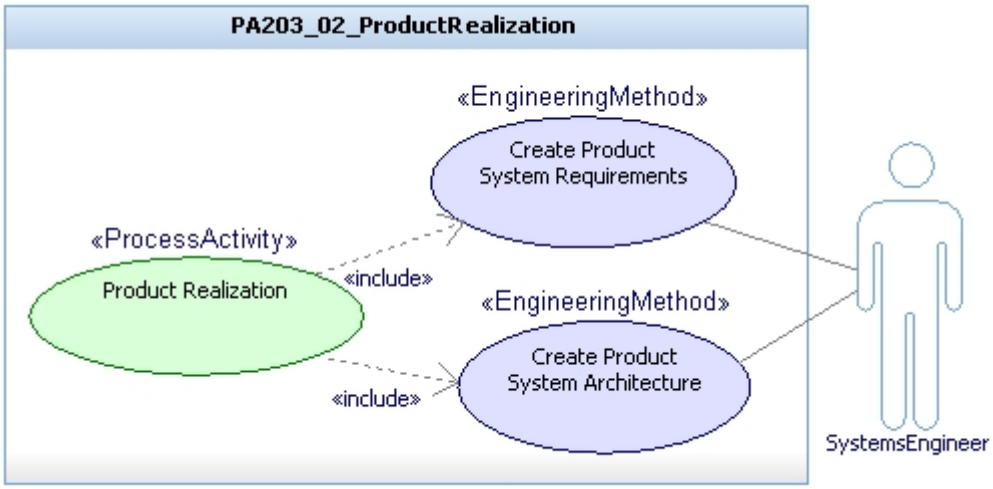

Figure 3-17 UCD_ProductRealization

3.2.2.2 EM203_02_01 - Create Product System Requirements

ID: EM203_02_01

Based on the product configuration the relevant system requirements for a product variant shall be identified and exported from the product family system requirements database into the product requirements documentation or database.
Variation points in the selected system requirements shall be decided for the product variant. The common and variable system requirements shall be identified for the product variant.

EM Purpose: The system requirements for a product variant shall be created consistently to functions, features.

EM Pre-Condition: Variation points and resolution rules are defined for functions and feature, system requirements of the product family. Functions, Features of the product variant are identified in a product variant description.

EM Post-Condition: System requirements variation points are resolved and system requirements are available for a product variant.

A transformation protocol documents the decisions leading to the product system requirements.

The product variant description is enhanced by the identified product system requirements.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. require list of product variant descriptions
02. create list of product variant descriptions
03. provide list of product variant descriptions
04. select a product variant description
05. provide a product variant description
06. identify system requirements

07. identify and resolve system requirements variation points
08. display the system requirements of the product variant
09. review the system requirements of the product variant
10. change feature selection for the product variant
11. store the requirements resolution
12. enhance product variant description with the requirements resolution
13. store the transformation protocol of the system requirements

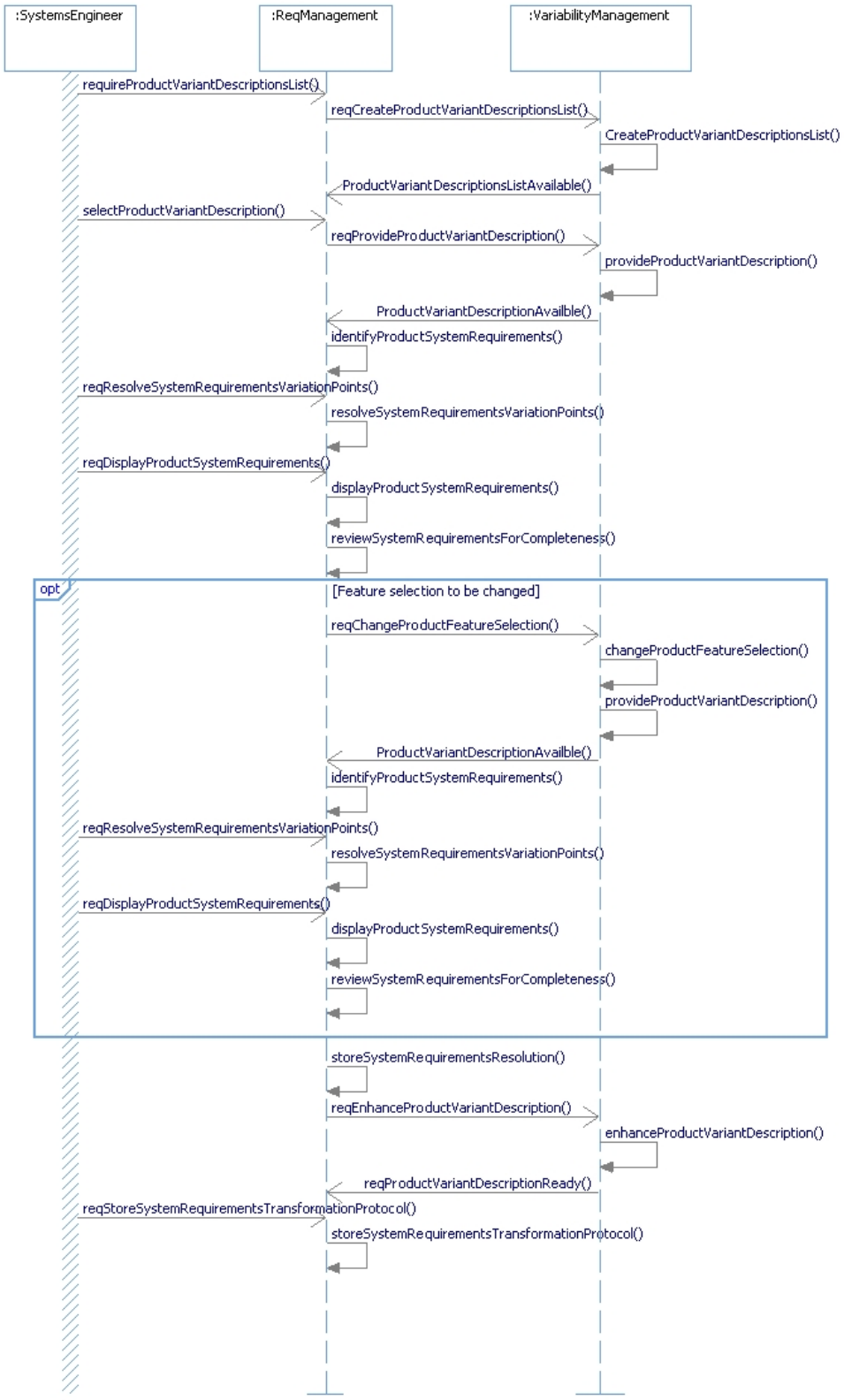

Figure 3-18 SD_EM203_02_01_ProductSystemRequirements

changeProductFeatureSelection: The feature selection needs to be adapted because of missing features, system requirements or system model elements.

Relates to Engineering Activity: 10. change product feature selection

CreateProductVariantDescriptionsList: A list of all available product variant descriptions shall be created.

Relates to Engineering Activity: 02. create list of product variant descriptions

displayProductSystemRequirements: The resolved product system requirements shall be displayed.

Relates to Engineering Activity: 08. display the system requirements of the product variant

enhanceProductVariantDescription: The product variant description shall be enhanced by the resolution of the system model variation points.

The selected system model options shall be stored in the product variant description.

Relates to Engineering Activity: 12. enhance product variant description by the product system model elements

identifyProductSystemRequirements: The system requirements for a product variant shall be identified according to the resolution rules using the selected features and feature options.

Relates to Engineering Activity: 06. identify system requirements

provideProductVariantDescription: The requested product variant description shall be provided.

Relates to Engineering Activity: 05. provide the product variant description

resolveSystemRequirementsVariationPoints: Each [variation point] within the system requirement having a binding time to be solved during construction e.g. compile and build time or integration time shall be resolved now by choosing the required options.

Relates to Engineering Activity: 07. identify and resolve system requirements variation points

reviewSystemRequirementsForCompleteness: The product system requirements shall be reviewed for completeness.

Relates to Engineering Activity: 09. review the system requirements of the product variant

storeSystemRequirementsResolution: The resulting system requirements for the product variant shall be stored as copy in the product realization environment.

Relates to Engineering Activity: 11. store the requirements resolution

storeSystemRequirementsTransformationProtocol: The used [Product Variant Description], the system requirements [Variation Point]s, the selected Options and the resolved requirements shall be documented in a [Transformation Protocol]. The [Transformation Protocol] shall describe by which decision (Common Requirement, Feature Option or variation point option) a system requirement was selected.

Relates to Engineering Activity: 13. store the transformation protocol of the system requirements

3.2.2.3 EM203_02_02 - Create Product System Architecture

ID: EM203_02_02

The product system architecture for a product variant shall be created.

The common system architecture artefacts shall be provided.
Only these architecture artefacts shall be provided which are required for the selected product variant.

EM Purpose: The system model for a product variant shall be created consistently to functions, features and system requirements.

EM Pre-Condition: Variation points and resolution rules are defined in the reference architecture of the product family. Functions, Features and System Requirements are identified in a product variant description.

EM Post-Condition: System model variation points are resolved and a system model is created for a product variant.

A transformation protocol documents the decisions leading to this product system model.

The product variant description is enhanced by the identified product system model elements.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. require list of product variant descriptions
02. create list of product variant descriptions
03. provide list of product variant descriptions
04. select a product variant description
05. provide the product variant description
06. identify the system model elements for the product variant
07. resolve system model variation points
08. display product system model
09. review system model for the product variant
10. change product feature selection
11. store resolution of system model variation points
12. enhance product variant description by the product system model elements
13. store transformation protocol of system model

Figure 3-19 SD_EM203_02_02_ProductSystemModelSynthesis

changeProductFeatureSelection: The feature selection needs to be adapted because of missing features, system requirements or system model elements.

Relates to Engineering Activity: 10. change product feature selection

CreateProductVariantDescriptionsList: A list of all available product variant descriptions shall be created.

Relates to Engineering Activity: 02. create list of product variant descriptions

createSystemModelTransformationProtocol: The used [Product Variant Description], the system model [Variation Point]s, the selected Options and the resolved system model elements shall be documented in a transformation protocol. The protocol shall describe by which decision (Common Element, Feature Option or VP option) a system model element was selected.

Relates to Engineering Activity: 13. store transformation protocol of system model

displayProductSystemModel: The resolved product system model shall be opened and displayed.

Relates to Engineering Activity: 08. display product system model

enhanceProductVariantDescription: The product variant description shall be enhanced by the resolution of the system model variation points.

The selected system model options shall be stored in the product variant description.

Relates to Engineering Activity: 12. enhance product variant description by the product system model elements

identifyProductSystemModelElements: The system model elements for a product variant shall be identified according the resolution rules using the selected features and feature options.

Relates to Engineering Activity: 06. identify the system model elements for the product variant

ProductVariantDecriptionsListAvailable: self-explanatory.

provideProductVariantDescription: The requested product variant description shall be provided.

Relates to Engineering Activity: 05. provide the product variant description

resolveSystemModelVariationPoints: Each [Variation Point] within the system model having a binding time to be solved during construction e.g. compile and build time or integration time shall be resolved now by choosing the required options.

Relates to Engineering Activity: 07. resolve system model variation points

reviewProductSystemModelForCompleteness: The system model for the product variant shall be reviewed if it is complete.

Relates to Engineering Activity: 09. review system model for the product variant

selectProductVariantDecription: A [Product Variant Description] description shall be selected.

3.3 User Story US204 - Ontology-based Requirements Engineering

This user story comprises an approach of requirements formalization using boilerplates, patterns and domain ontologies in order to improve the quality of requirements:

- Develop domain ontology by defining concepts, axioms and relations that hold in the application domain
- Define boilerplates by defining sequences of fixed syntax elements and attributes for each type of boilerplate to cover the relevant requirement statements
- Capture requirements by selecting boilerplates and filling in of boilerplates using guidance of domain ontology
- Validate and update domain ontology by changing concepts, axioms and relations of the domain ontology to improve guidance and analysis results
- Define patterns by guided formalization of boilerplates to enable advanced formal analysis techniques
- Analyze the quality of requirements set by computing quality properties using ontology-based and pattern-based analysis
- Interpret the analysis results and improve requirements
- Manage the traceability between requirements and related artefacts.

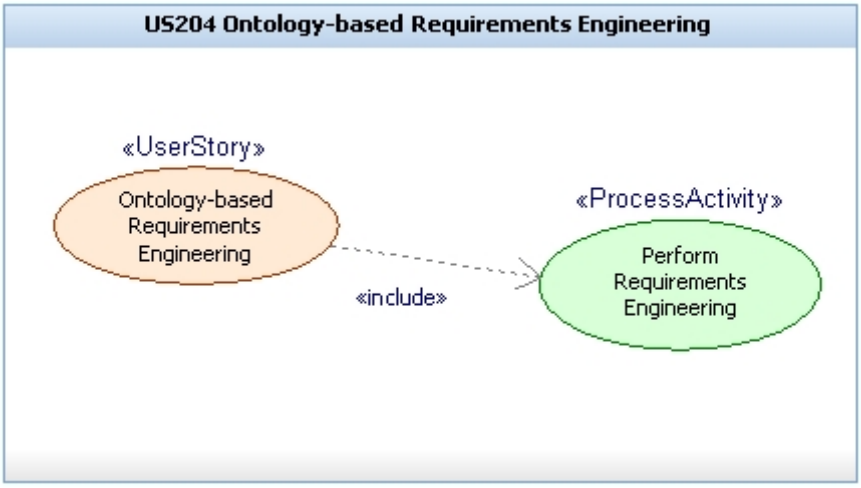

Figure 3-20 UCD_US204_OntologyBasedRequirementsEngineeringPkg

The User Story Ontology-based Requirements Engineering is including the Process Activities listed below. Note, that a Process Activity is detailed within another User Story if no <<include>> dependency is established from the User Story.

- Perform Requirements Engineering

3.3.1 PA204_01 - Perform Requirements Engineering

ID: PA204_01

Related User Story: Ontology-based Requirements Engineering

The process activity "Perform Requirements Engineering" describes the systems engineering activities related to the elicitation, development and analysis of stakeholder and system requirements. The subsequent description is split into two sub-activities
- Define Stakeholder Needs and Requirements
- Define System Requirements.

The purpose of "Define Stakeholder Needs and Requirements" is to define the stakeholder requirements for a system that can provide the capabilities needed by users and other stakeholders in a defined environment. It involves eliciting the needs from identified stakeholders, analyzing and transforming these needs into a common set of stakeholder requirements which shall be recorded in a form suitable for requirements management throughout the life cycle. These records establish the stakeholder requirements baseline. Stakeholder requirements are the basis for traceability to stakeholder needs, system requirements, and subsequent system elements.

The purpose of "Define System Requirements" (see below) is to transform the stakeholder, user-oriented view of desired capabilities into a technical view of a solution that meets the operational needs of the user. It involves creating a set of verifiable system requirements that specify what characteristics, attributes, and functional and performance requirements the system is to possess, in order to satisfy stakeholder requirements. System requirements shall be recorded in a form suitable for requirements management throughout the life cycle. These records establish the system requirements baseline. System requirements are the basis for traceability to stakeholder requirements and subsequent system elements.

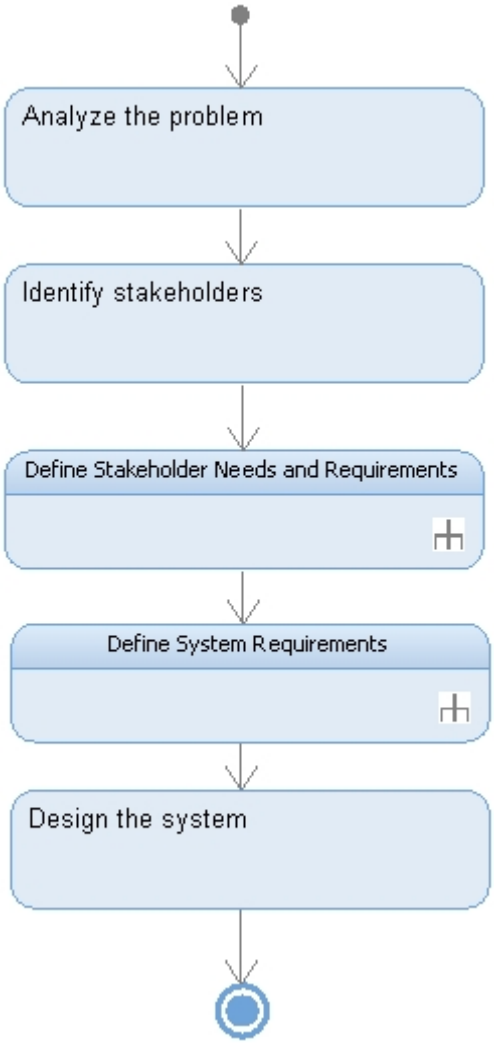

Figure 3-21 AD_PA204_01

Analyze the problem: Understand the need for the system. Determine the vision, the scope of the system, and its boundaries. Identify the key functionalities. Decide which functionalities are most critical. Assure that at least one possible solution exists. Understand the costs, schedule, and risks associated with the project.

Design the system: The main objectives of the system design process are to determine how to partition the system and to identify which requirements should be allocated to which system elements. As system elements are defined, additional requirements statements should be created to define relationships between the architectural elements of the system, to provide necessary clarity in the context of the lower levels of abstraction of the system elements, or to specify design constraints or performance figures for system elements. This process may be applied recursively through several successive levels of system elements within the system structure down to software and hardware requirements which can be implemented by software and hardware teams.

Identify stakeholders: Identify the stakeholders who have a valuable interest in the system. The stakeholders are the primary source from who needs and requirements should be elicited. Some examples of stakeholders are:

- customer

- users and operators

- maintainers

- regulatory agencies and authorities

The role of stakeholders in relation with the system should be clarified and documented.

Define Stakeholder Needs and Requirements: This action is a sub activity flow of the main activity diagram "AD_PA204_01" and is described by the Activity Diagram below.

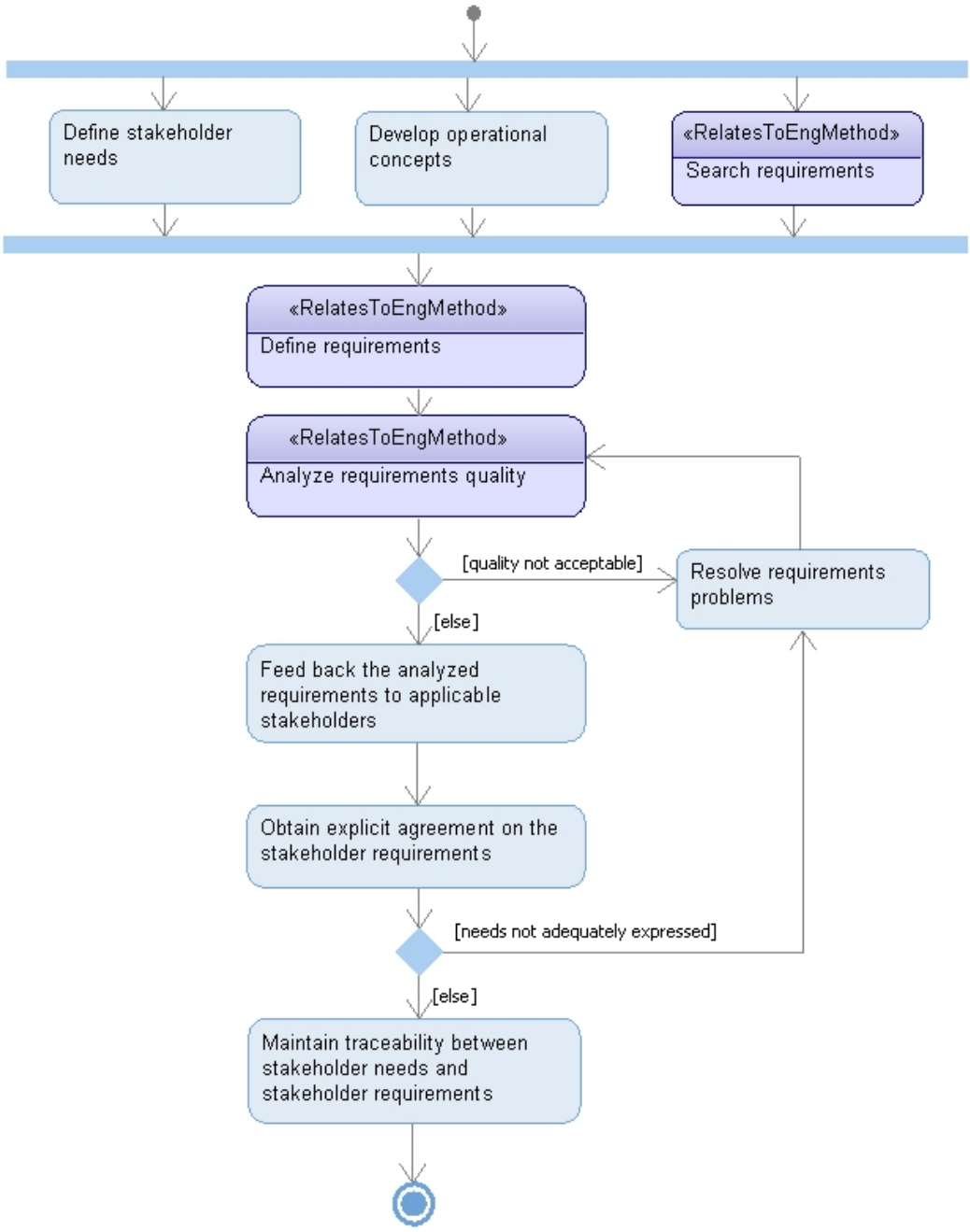

Figure 3-22 Define Stakeholder Needs and Requirements

Analyze requirements quality (RelatesToEngMethod): Stakeholder requirements should be analyzed for characteristics of individual requirements, as well as characteristics of the set of requirements. Potential analysis characteristics include that the requirements are necessary, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, verifiable, affordable, and bounded.

Define requirements (RelatesToEngMethod): Define stakeholder requirements consistent with life-cycle concepts, scenarios, interactions, solution constraints, and critical qualities such as assurance, safety, security, environment, or health.

Define stakeholder needs: This activity consists of the following tasks:

- identify the stakeholders who have an interest in the system throughout its life cycle

- elicit stakeholder needs from the identified stakeholders

- prioritize and down-select needs

- define the stakeholder needs

Develop operational concepts: This activity consists of the following tasks:

- define context of use within the concept of operations

- define a representative set of scenarios to identify all required capabilities that correspond to anticipated operational concept

- identify the interaction between users and the system

Feed back the analyzed requirements to applicable stakeholders: Ensure that the needs and expectations have been adequately captured and expressed.

Maintain traceability between stakeholder needs and stakeholder requirements: Initial requirements traceability should be established and maintained to document how the formal requirements are intended to meet the stakeholder objectives and achieve stakeholder agreement.

Obtain explicit agreement on the stakeholder requirements: This includes confirming that stakeholder requirements are comprehensible to originators and that the resolution of conflict in the requirements has not corrupted or compromised stakeholder intentions.

Resolve requirements problems: In case that the requirements quality is not acceptable, requirements need to be updated. It is important to continue to perform requirements negotiation during the analysis and allocation of requirements. Negotiation might be needed among stakeholders requiring mutually incompatible features, or due to conflicts between desired performance requirements, constraints, available budget, and delivery schedule.

Search requirements (RelatesToEngMethod): Identify and assess opportunities to re-use previously existing requirements. This includes identification of existing systems that provide similar functions or capabilities, specified functions or capabilities applicable to the new system-of-interest, and information on the extent of reusability.

Define System Requirements: This action is a sub activity flow of the main activity diagram "AD_PA204_01" and is described by the Activity Diagram below.

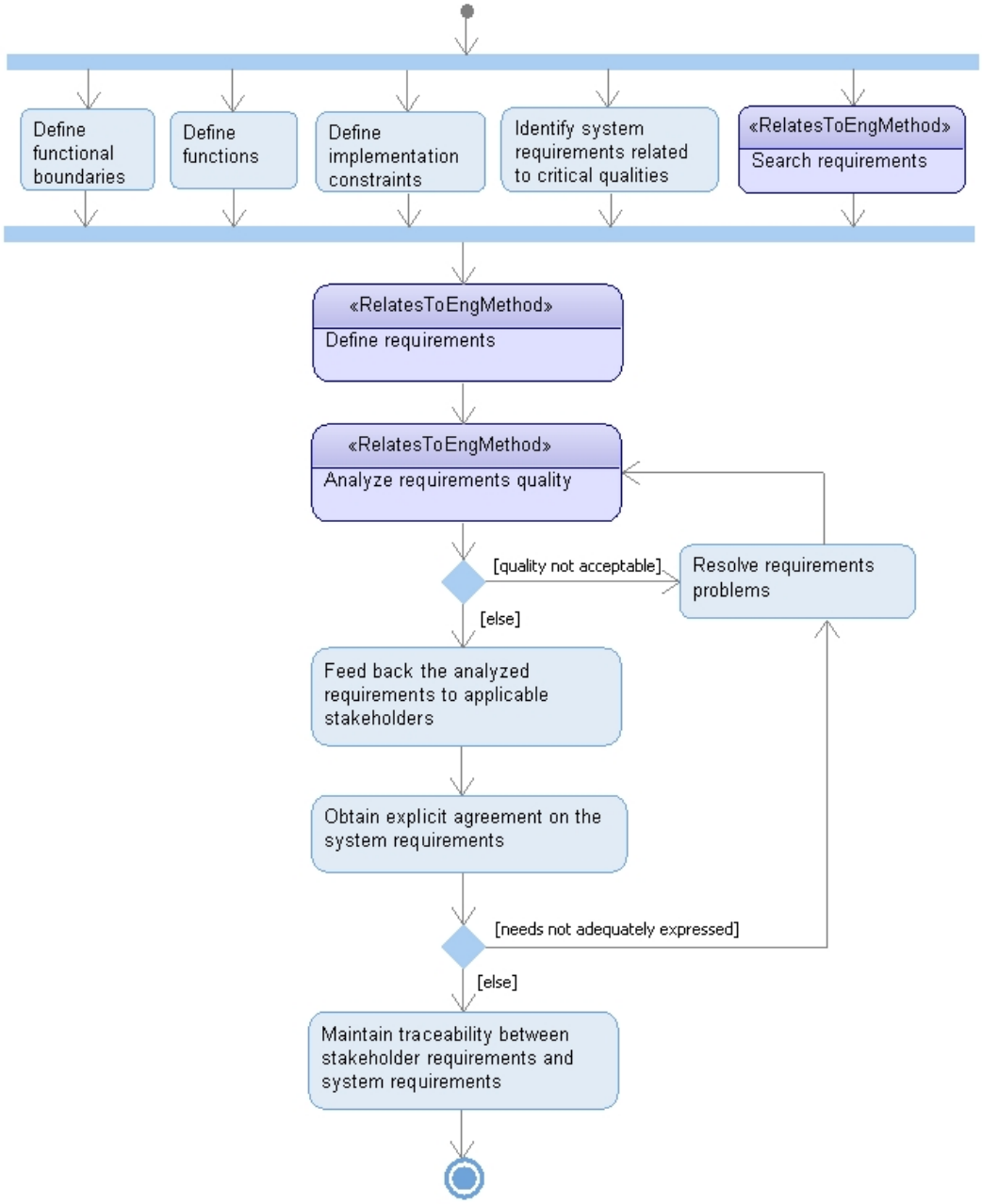

Figure 3-23 Define System Requirements

Analyze requirements quality (RelatesToEngMethod): System requirements should be analyzed for characteristics of individual requirements, as well as characteristics of the set of requirements. Potential analysis characteristics include that the requirements are necessary, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, verifiable, affordable, and bounded.

Define functional boundaries: Define the functional boundary of the system in terms of the behaviour and properties to be provided. This includes the system's stimuli and its responses to user and environment behaviour, and an analysis and description of the required interactions between the system and its operational environment in terms of interface constraints, such as mechanical, electrical, mass, thermal, data, and procedural flows. This establishes the expected system behaviour, expressed in quantitative terms, at its boundary.

Define functions: Define each function that the system is required to perform. This includes how well the system is required to perform that function, the conditions under which the system is to be capable of performing the function. Conditions for the performance of functions may incorporate reference to required states and modes of operation of the system. System requirements depend heavily on abstract representations of proposed system characteristics and may employ multiple modelling techniques and perspectives to give a sufficiently complete description of the desired system requirements.

Define implementation constraints: Define necessary implementation constraints that are introduced by stakeholder requirements or are unavoidable solution limitations. This includes the implementation decisions that are allocated from design at higher levels in the structure of the system.

Define requirements (RelatesToEngMethod): Define system requirements consistent with stakeholder requirements, functional boundaries, functions, implementation constraints, and critical quality characteristics.

Feed back the analyzed requirements to applicable stakeholders: Ensure that the needs and expectations have been adequately captured and expressed.

Identify system requirements related to critical qualities: This includes defining critical performance parameters associated with each effectiveness measure identified in the stakeholder requirements. The critical performance measures are analyzed and reviewed to ensure stakeholder requirements are met and to ensure identification of project cost, schedule or performance risk associated with any non-compliance.

Maintain traceability between stakeholder requirements and system requirements: Initial requirements traceability should be established and maintained to document how the formal requirements are intended to meet the stakeholder objectives and achieve stakeholder agreement.

Obtain explicit agreement on the system requirements: This includes confirming that system requirements are comprehensible to originators and that the resolution of conflict in the requirements has not corrupted or compromised stakeholder intentions.

Resolve requirements problems: In case that the requirements quality is not acceptable, requirements need to be updated. It is important to continue to perform requirements negotiation during the analysis and allocation of requirements. Negotiation might be needed among stakeholders requiring mutually incompatible features, or due to conflicts between desired performance requirements, constraints, available budget, and delivery schedule.

Search requirements (RelatesToEngMethod): Identify and assess opportunities to re-use previously existing requirements. This includes identification of existing systems that provide similar functions or capabilities, specified functions or capabilities applicable to the new system-of-interest, and information on the extent of reusability.

3.3.1.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

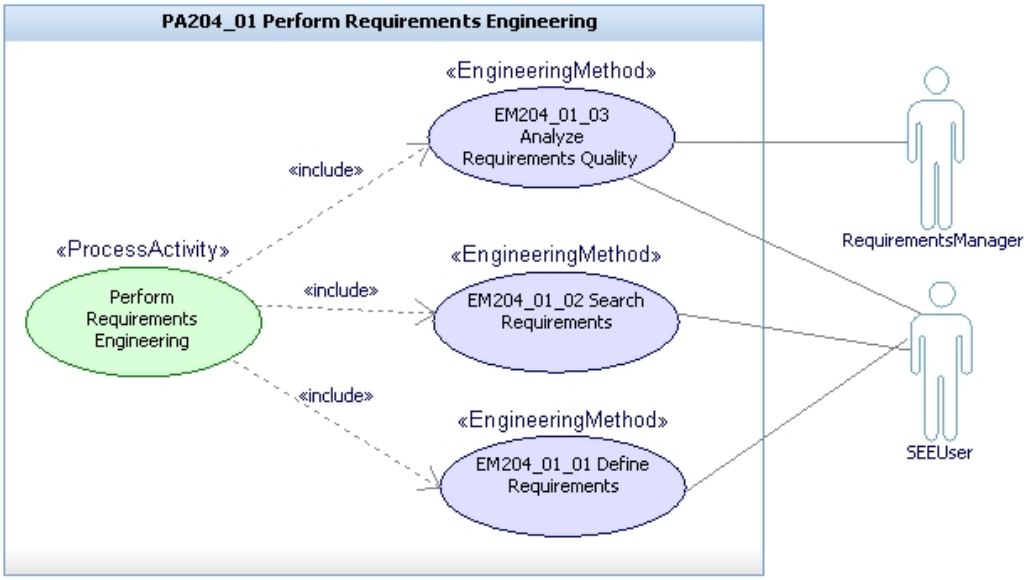

Figure 3-24 UCD_PerformRequirementsEngineering

3.3.1.2 EM204_01_01 - Define Requirements

ID: EM204_01_01

During requirements writing, natural-language statements are matched with boilerplates and concepts from domain ontology in order to give the SEEUser immediate feedback on requirements quality issues and propose resolutions.

EM Purpose: Support the requirements author during writing by requirements quality evaluation.

EM Pre-Condition: Requirement is stored in the requirements management database. Initially, the requirement may be empty.

EM Post-Condition: Improved requirement is stored in the requirements management database.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Select a requirement for editing
02. Retrieve the selected requirement from the requirements management database
03. Display the selected requirement
04. Retrieve the boilerplates of the boilerplate group assigned to the current requirements module
05. Match the current requirement with the boilerplates in order to find the best fitting boilerplates
06. Propose terms of the controlled vocabulary fitting with the matched boilerplates
07. Analyze requirements quality and create findings in order to guide improvement
08. Display requirements quality findings
09. Improve the requirement based on requirements quality findings
10. Update the requirement in the requirements management database.

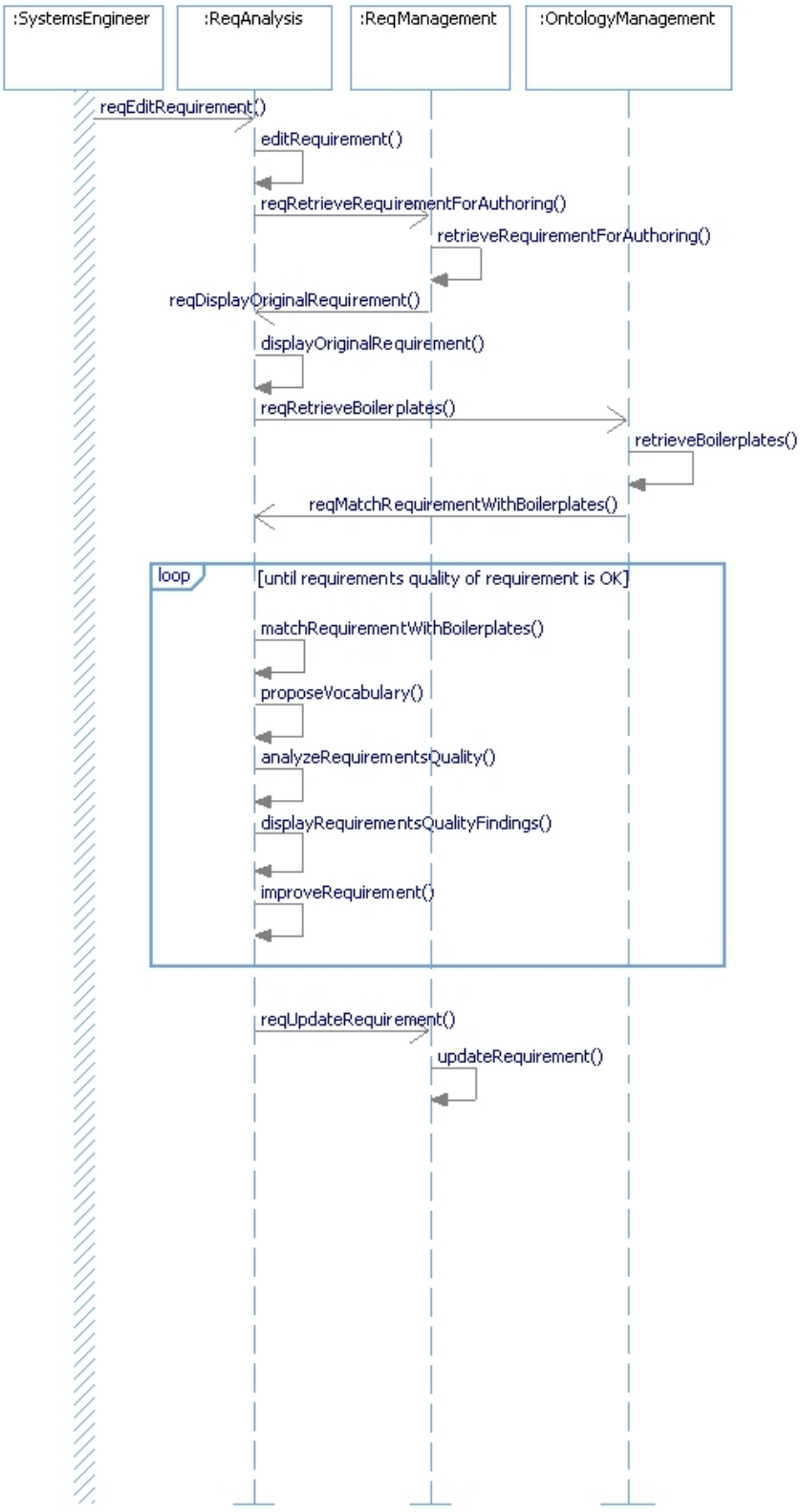

Figure 3-25 SD_EM204_01_01_Define Requirements

analyzeRequirementsQuality: Quality metrics are calculated for selected requirements based on current boilerplate and quality metrics groups. In addition, findings are created during requirements quality evaluation which guide the improvement of requirements.

Relates to Engineering Activity: 07. Analyze requirements quality and create findings in order to guide improvement

displayOriginalRequirement: Display the original requirement text.

Relates to Engineering Activity: 03. Display the selected requirement

displayRequirementsQualityFindings: Display the findings of the requirements quality analysis in order to provide hints for requirements improvement.

Relates to Engineering Activity: 08. Display requirements quality findings

editRequirement: Select a requirement from the requirements management database for editing.

Relates to Engineering Activity: 01. Select a requirement for editing

improveRequirement: Improve the requirement based on the calculated quality metrics and the findings of the quality analysis.

Relates to Engineering Activity: 09. Improve the requirement based on requirements quality findings

matchRequirementWithBoilerplates: Match the currently edited requirement with the boilerplates of the current boilerplate group in order to find the best fitting boilerplates.

Relates to Engineering Activity: 05. Match the current requirement with the boilerplates in order to find the best fitting boilerplates

proposeVocabulary: Propose terms of the controlled vocabulary defined in the domain ontology fitting with the matched boilerplates.

Relates to Engineering Activity: 06. Propose terms of the controlled vocabulary fitting with the matched boilerplates

retrieveBoilerplates: Retrieve the boilerplates of the boilerplate group assigned to the current requirements module.

Relates to Engineering Activity: 04. Retrieve the boilerplates of the boilerplate group assigned to the current requirements module

retrieveRequirementForAuthoring: For the selected requirement retrieve the requirements ID, the requirements text, the version history and optionally pre-defined attribute values.

Relates to Engineering Activity: 02. Retrieve the selected requirement from the requirements management database

updateRequirement: Updates the given requirement in the requirements management database.

Relates to Engineering Activity: 10. Update the requirement in the requirements management database

3.3.1.3 EM204_01_02 - Search Requirements

ID: EM204_01_02

Construct queries to search for requirements artefacts, e.g. similar requirements for re-use, requirements of a given type (e.g. functional, performance, safety) or requirements related to a given concept.

EM Purpose: Support the requirements engineering in reusing previously defined requirements.

EM Pre-Condition: Requirements are stored in the requirements management database.

EM Post-Condition: Formalized requirements are stored in knowledge base for search and retrieval.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Formalize natural language statement into knowledge structures that support reasoning and retrieval
02. Store knowledge structure
03. Define search criterion.
04. Retrieve knowledge structures including references to requirements matching the search criterion
05. Retrieve referenced requirements from requirements management database
06. Select requirements for analysis

Figure 3-26 SD_EM204_01_02_Retrieve Requirements

defineSearchCriterion: Define a search criterion for finding matching requirements. Semantic search criteria can be based on requirements type, concepts occurring in requirements, or similarity with existing requirements.

Relates to Engineering Activity: 03. Define search criterion.

retrieveKnowledgeStructure: Retrieve all knowledge structures including references to the formalized requirements matching with a given search criteria from the knowledge base.

Relates to Engineering Activity: 04. Retrieve knowledge structures including references to requirements matching the search criterion

retrieveRequirements: For each requirement of the selected requirements group retrieve the requirements ID, the requirements text, the version history and optionally pre-defined attribute values.

Relates to Engineering Activity: 05. Retrieve referenced requirements from requirements management database

selectRequirements: Select the requirements for further investigation.

Relates to Engineering Activity: 06. Select requirements for analysis

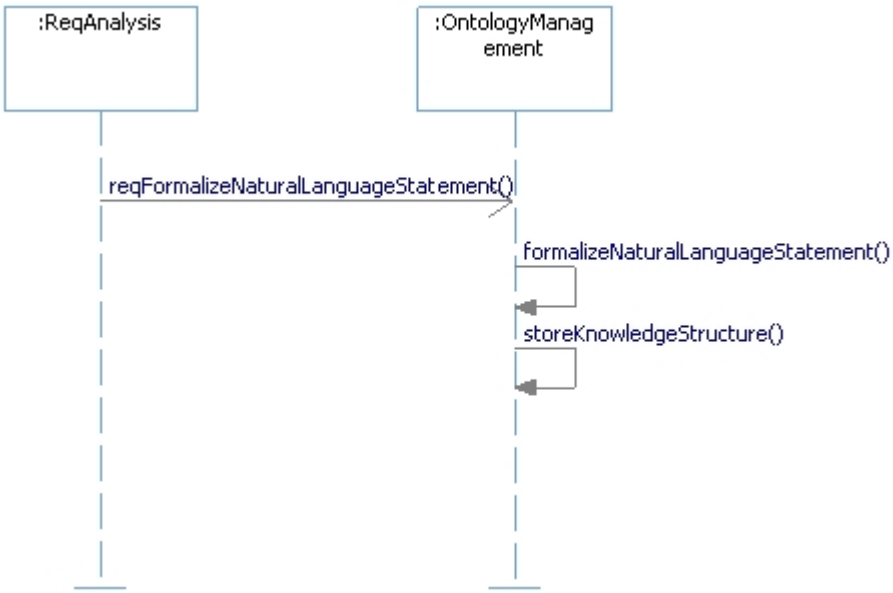

Figure 3-27 SD_EM204_01_02_Formalize Requirements

formalizeNaturalLanguageStatement: Perform the natural language processing process including tokenization, normalization, tagging, semantic grouping, pattern matching and formalization.

Relates to Engineering Activity: 01. Formalize natural language statement into knowledge structures that support reasoning and retrieval

storeKnowledgeStructure: Store the knowledge structures of the formalized requirement including a reference to the requirement in the knowledge base.

Relates to Engineering Activity: 02. Store knowledge structure

3.3.1.4 EM204_01_03 - Analyze Requirements Quality

ID: EM204_01_03

Analyze the quality of a given set of natural-language requirements according to a configured set of quality metrics, boilerplate group and domain ontology with the objective to improve the requirements quality.

EM Purpose: Improve requirements quality by requirements quality evaluation.

EM Pre-Condition: Requirements are stored in the requirements management database.

EM Post-Condition: Improved requirements are stored in the requirements management database including links to quality evaluation results.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Define quality metrics
02. Select requirements module
03. Assign quality metrics to requirements module
04. Analyze requirements module
05. Retrieve requirements for analysis
06. Select requirements for analysis
07. Analyze requirements quality and create findings in order to guide improvement
08. Create requirements quality report
09. Improve requirements based on requirements quality findings
10. Update requirements in the requirements management database

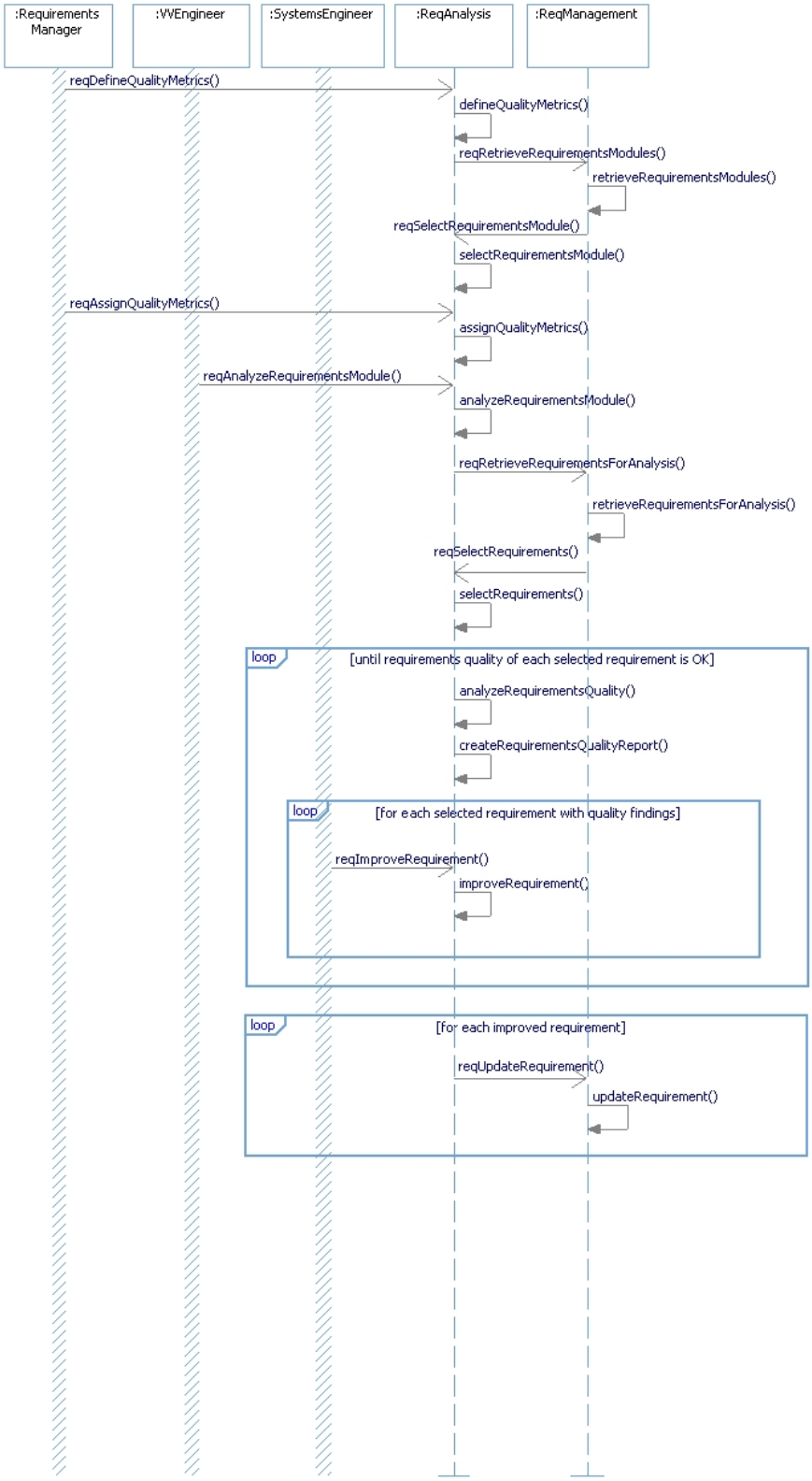

Figure 3-28 SD_EM204_01_03_Analyze Requirements Quality

analyzeRequirementsModule: Perform quality evaluation of a given requirements group.

Relates to Engineering Activity: 04. Analyze requirements module

analyzeRequirementsQuality: Quality metrics are calculated for selected requirements based on current boilerplate and quality metrics groups. In addition, findings are created during requirements quality evaluation which guide the improvement of requirements.

Relates to Engineering Activity: 07. Analyze requirements quality and create findings in order to guide improvement

assignQualityMetrics: For the selected requirements group, assign the quality metrics group which shall be applied for quality evaluation. If boilerplate related metrics shall be applied, e.g. boilerplate matching of requirements or completeness check, the boilerplate group that shall be applied for an analysis run, has to be assigned in addition.

Relates to Engineering Activity: 03. Assign quality metrics to requirements module

createRequirementsQualityReport: Create a quality report that contains values and comments for the individual metrics in order to guide the improvement of the analyzed set of requirements.

Relates to Engineering Activity: 08. Create requirements quality report

defineQualityMetrics: Select the metrics which shall be used for quality evaluation. Priorities (weight functions) can be assigned to distinguish between important and less important metrics. Define for each chosen metric the quality function, i.e. define the value ranges with acceptable or not acceptable quality, respectively.

Relates to Engineering Activity: 01. Define quality metrics

improveRequirement: Improve the requirement based on the calculated quality metrics and the findings of the quality analysis.

Relates to Engineering Activity: 09. Improve the requirement based on requirements quality findings

retrieveRequirementsForAnalysis: For each requirement of the selected requirements group retrieve the requirements ID, the requirements text, the version history and optionally pre-defined attribute values.

Relates to Engineering Activity: 05. Retrieve requirements for analysis

retrieveRequirementsModules: Retrieve all requirements modules contained in the requirements management database.

Relates to Engineering Activity: 02. Select requirements module

selectRequirements: Select the requirements for further investigation.

Relates to Engineering Activity: 06. Select requirements for analysis

selectRequirementsModule: Select the requirements group which shall be analyzed.

Relates to Engineering Activity: 02. Select requirements module

updateRequirement: Updates the given requirement in the requirements management database.

Relates to Engineering Activity: 10. Update the requirement in the requirements management database

3.4 User Story US205 - Process Automation, Guidance and Monitoring

The user story “Process Automation, Guidance and Monitoring" concerns:

- The automation of standard tasks throughout the complete system engineering environment (SEE). This includes the integration of the tools in a way relieving the user from repeating similar tasks for each tool (e.g. authentication, saving data, exporting data).
- Company specific change management and change control for all the artefacts to be managed in the SEE. Change control is applied to all configuration items including data base elements, binary data, documents and other work products.
- Company specific implementation of workflows (e.g. change request) while controlling the assigned artefacts for the user commissioned for a dedicated task.
- Generation of different product views by defining the scope of different baseline sets. Product views also allow the tailoring of deliveries e.g. to subordinated development teams and external suppliers.
- Automated collection of data throughout the SEE tools and generation of related reports.

The user story "Process Automation, Guidance and Monitoring" includes integral processes that are applied continuously throughout the life cycle.

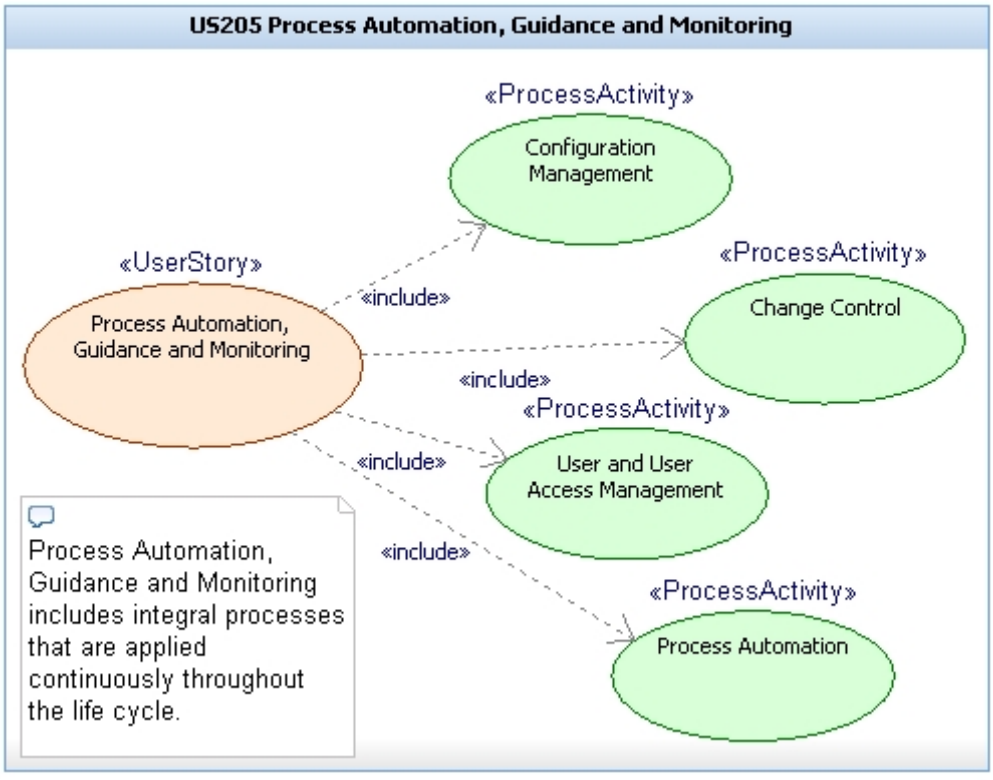

Figure 3-29 UCD_US205_ProcessAutomation

The User Story Process Automation, Guidance and Monitoring is including the Process Activities listed below. Note, that a Process Activity is detailed within another User Story if no <<include>> dependency is established from the User Story.

- Configuration Management
- Change Control
- User and User Access Management
- Process Automation

3.4.1 PA205_01 - Configuration Management

ID: PA205_01

Related User Story: Process Automation, Guidance and Monitoring

Configuration Management (CM) is a process for establishing and maintaining consistency of a product's performance, functional and physical attributes with its requirements, design and operational information throughout its life.

Besides Change Control which is tackled in a separate process activity, configuration management includes baselining and maintenance of a revision history. While change control manages the changes to configuration items by controlling read / write access to the artefacts, the other activities log and report the changes, which are incorporated for a dedicated baseline allowing version management.

A major issue for version management is branching and merging. The challenge for these two associated activities is the IOS basic pattern that each tool hosts its own data. Furthermore, one need to consider that the data is hosted in a database and not in central version management tool as it is common for SW development.

Following constraints need to be considered:

- The tools hosting a database need to support version management. As every tool needs to implement the functionality only a simple functionality should be required. A more complex functionality is difficult to harmonize throughout the different tools and is expensive to implement.
- Tools should concentrate on their core functionality. Especially tools provided by academia or later on by the users themselves should not be constraint by requirements not related to the provided functionality. A complex version management is an issue for tool providers specializing in this domain. Therefore, the functionality a single tool needs to provide should be simple.
- Version management with branching and merging is complicated to implement for tools hosting a database.

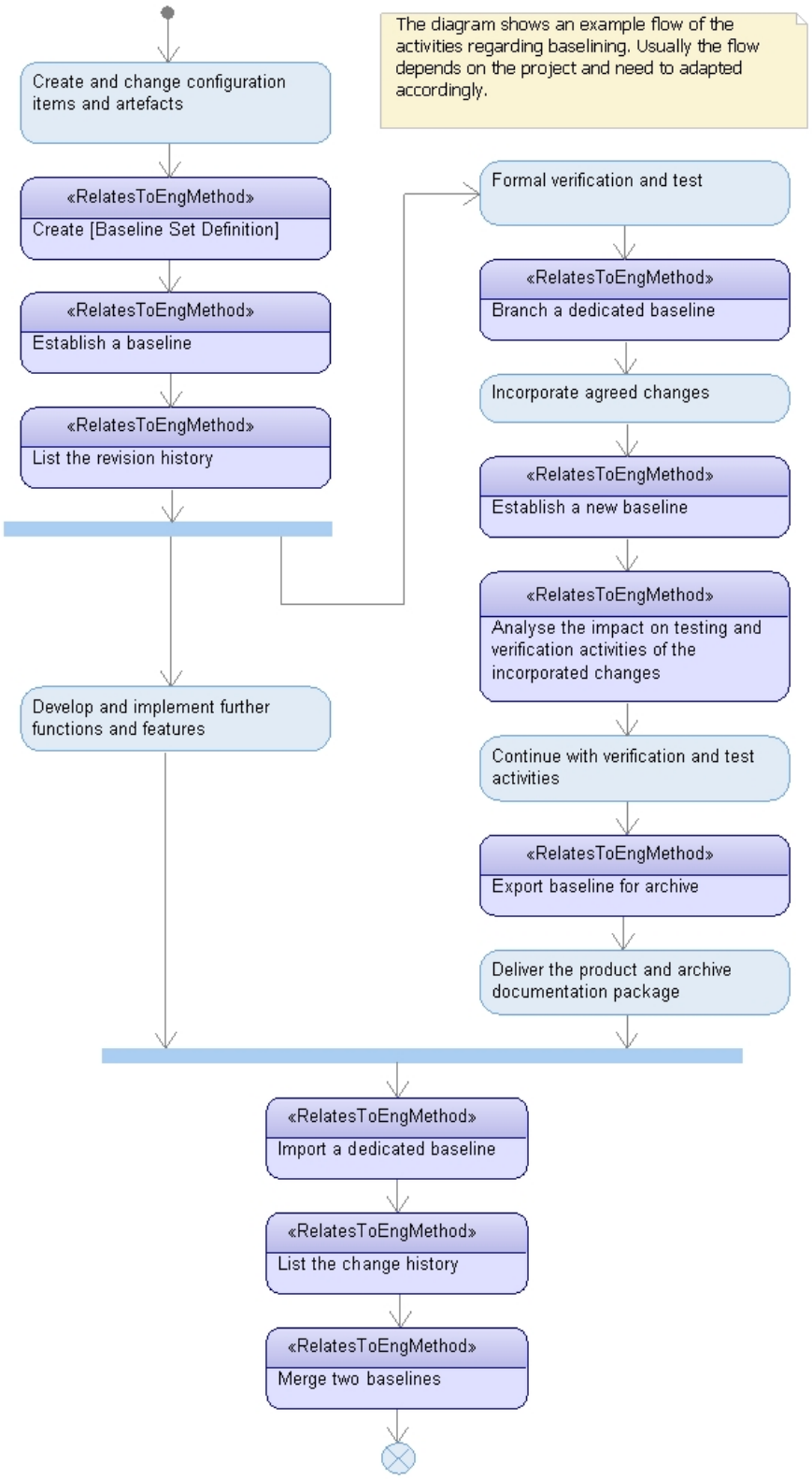

Figure 3-30 AD_PA205_01

Analyse the impact on testing and verification activities of the incorporated changes (RelatesToEngMethod): A change impact analysis may be performed to identify the test and verification cases and associated activities that are affected by a change and need to be repeated.

Branch a dedicated baseline (RelatesToEngMethod): If during formal test and verification failures are detected, there are two possibilities. Assumed that there are some minor problems, one option is to fix the detected errors in the baseline and to continue with testing and verification.

Therefore, you need to create a new branch based on an available baseline set under test.

Continue with verification and test activities: Repeat activities affected by the incorporated changes.

Create [Baseline Set Definition] (RelatesToEngMethod): The first step for baselining is to define a [Baseline Set Definition] with a dedicated scope. This baseline set refers to all the configuration items necessary for a dedicated scope. Any other configuration items, necessary to get a complete working environment should be included, so that at any point the SEE user is able to regenerate a working environment from those items.

The [Baseline Set Definition] is a subset of all the configuration items available in the SEE environment serving a dedicated purpose or scope.

Different baseline sets may be established as a tool's data may be part of different scopes.

Create and change configuration items and artefacts: During initial development many configuration items are generated and changed. Usually there is no or only very limited change control active during this phase. This phase continues until a certain maturity is gained. System elements are engineering tested and integrated.

Having a sufficient maturity gained the next phase could be envisaged.

Deliver the product and archive documentation package: After successful verification and test the product is delivered to the customers.

Develop and implement further functions and features: After drawing the baselines, the development teams usually continue to implement further features.

Establish a baseline (RelatesToEngMethod): A second step is to baseline the identified configuration items and artefacts. In its simpliest form this is to archive a copy of the configuration items listed in the baseline set definition. Each baseline has its unique identification.

Establish a new baseline (RelatesToEngMethod): establish a new (usually minor) baseline containing the agreed bug fixes and changes.

Export baseline for archive (RelatesToEngMethod): The baseline successfully tested is archived in PLM before it is delivered to customers.

Formal verification and test: Formal test, integration and verification is not started before a baseline is drawn assuring the repeatability of these activities. The formal verification and test is performed in parallel to development activities.

Import a dedicated baseline (RelatesToEngMethod): The baseline is imported from the archive to identify the changes made.

Incorporate agreed changes: Based on an approved change request (see PA205_02 Change Control) changes and bug fixes are incorporated.

List the change history (RelatesToEngMethod): List the changes made to the configuration items including

the change requests that are fixed

the configuration items that are changed

what was changed in the configuration items

List the revision history (RelatesToEngMethod): The consistency of all the configuration items is supported by listing all the baseline identifications for the baseline set. Using this list, you are able to check which items have change compared to a prior list.

Merge two baselines (RelatesToEngMethod): After successful verification and test the identified and implemented changes can be merged with the development trunk. The changes are then available for the next functional baseline.

3.4.1.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

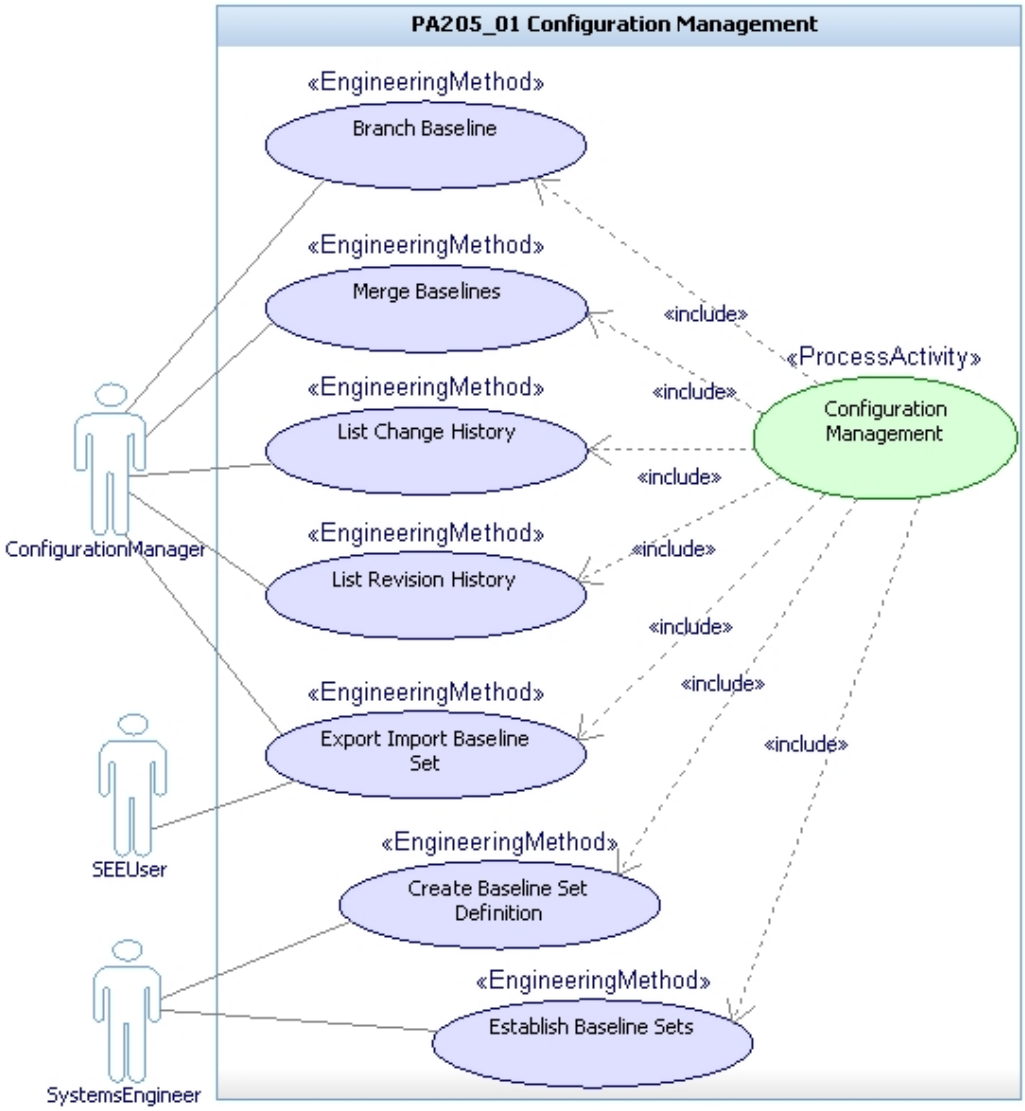

Figure 3-31 UCD_ConfigurationManagement

3.4.1.2 EM205_01_01 - Create Baseline Set Definition

ID: EM205_01_01

Define a Baseline Set Definition with a dedicated scope. Include all the configuration items necessary for that scope and other configuration items necessary.
The Baseline Set Definition is a subset of all the configuration items available in the SEE environment serving a dedicated purpose or scope.
Different baseline sets may be established as a tool's data may be part of different scopes.

EM Purpose: Generate a baseline set definition as a template for the baseline sets.

EM Pre-Condition: Project file and configuration items are available.

EM Post-Condition: Baseline Set Definition generated containing all the configuration items of interest for a dedicated purpose, scope or delivery.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Create the Baseline Set Definition List and request a list of config items from each tool.
02. The Tool provides the list of all its config items.
03. For each tool select the config items from the list provided.
04. Store relevant information in the baseline set definition list for each configuration item.
05. For each configuration item the dependencies are requested from the hosting tool.
06. The tool provides for the given configuration item all dependencies to other configuration items.
07. The SEE removes from the received list all configuration items already considered in the definition.
08. The user selects from the configuration items not yet considered those he wants to add to the baseline set definition.

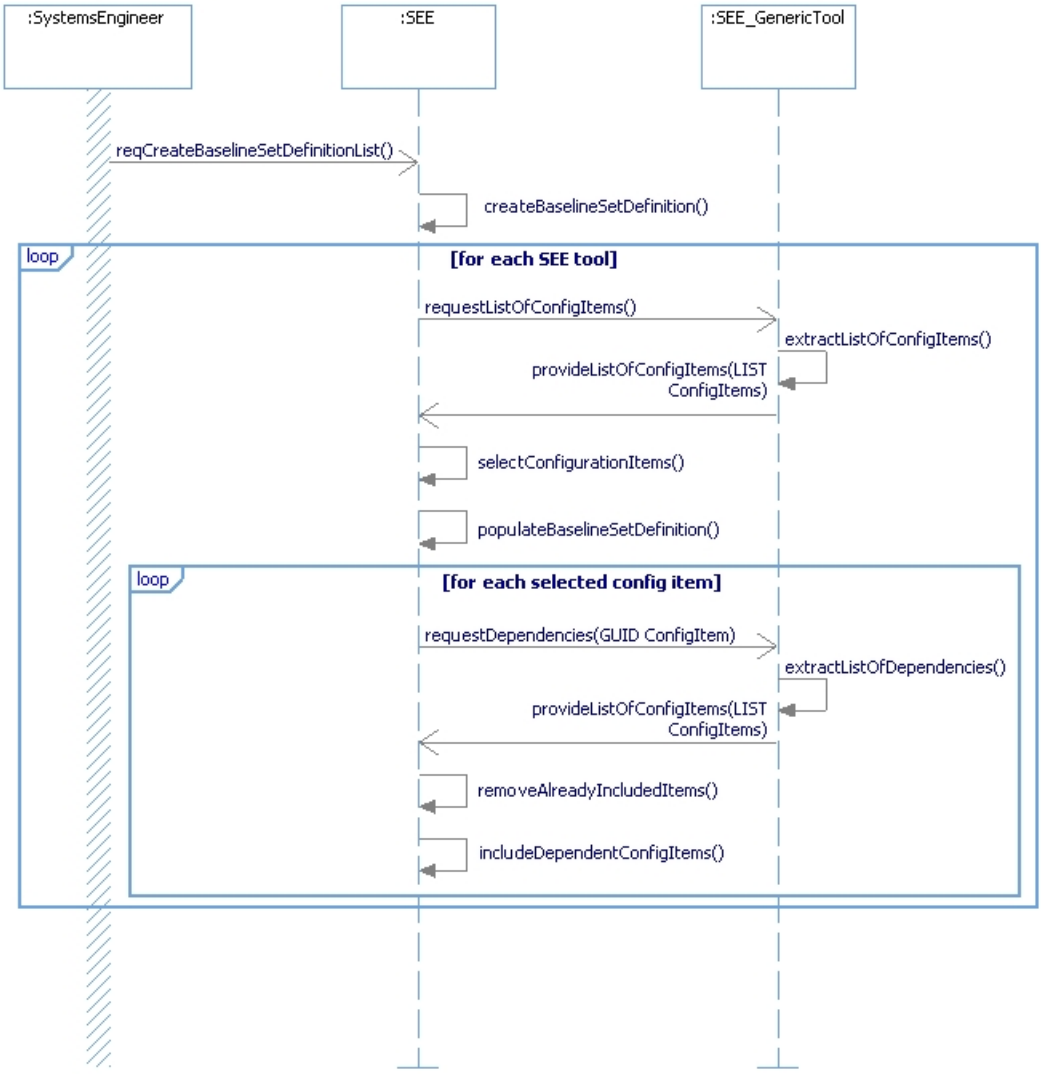

Figure 3-32 SD_EM205_01_01_CreateBaselineSetDefinition

createBaselineSetDefinition: Define the tool data and artefacts managed within the addressed tool to be included in a baseline. Different baseline sets may be established as a tool's data may be part of different scopes.

Relates to Engineering Activity: 01. Create the Baseline Set Definition List and request a list of config items from each tool.

extractListOfConfigItems: Extracts a complete set of configuration items included in the currently loaded database.

Relates to Engineering Activity: 04. The tool provides a list of all available configuration items.

extractListOfDependencies: Extracts for the given configuration item the dependencies to other configuration items.

Relates to Engineering Activity: 06. The tool provides for the given configuration item all dependencies to other configuration items.

includeDependentConfigItems: The user selects from the remaining configuration items not yet considered those he wants to add to the baseline set definition.

Relates to Engineering Activity: 08. The user selects from the configuration items not yet considered those he wants to add to the baseline set definition.

populateBaselineSetDefinition: The list is populated with the information of the selected configuration items like source tool, GUID, full name of the configuration item probably including a path.

Relates to Engineering Activity: 04. Store relevant information in the baseline set definition list for each configuration item.

removeAlreadyIncludedItems: The SEE removes all configuration items from the received list, that are already considered.

Relates to Engineering Activity: 07. The SEE removes from the received list all configuration items already considered in the definition.

selectConfigurationItems: For each tool select the config items from the list provided.

Relates to Engineering Activity: 03. For each tool select the config items from the list provided.

3.4.1.3 EM205_01_02 - Branch Baseline

ID: EM205_01_02

Create a new branch based on an available baseline set.

EM Purpose: To generate a new branch or working stream.

EM Pre-Condition: A trunk or a parent branch is available with baseline sets established.

EM Post-Condition: A new branch is created.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The configuration manager requests to generate a branch from any baseline set.

02. At the SEE the user selects one baseline set from the available baseline sets. For each configuration item the SEE asks the tool if it supports the history feature.

03. Show the status of the complete operation.

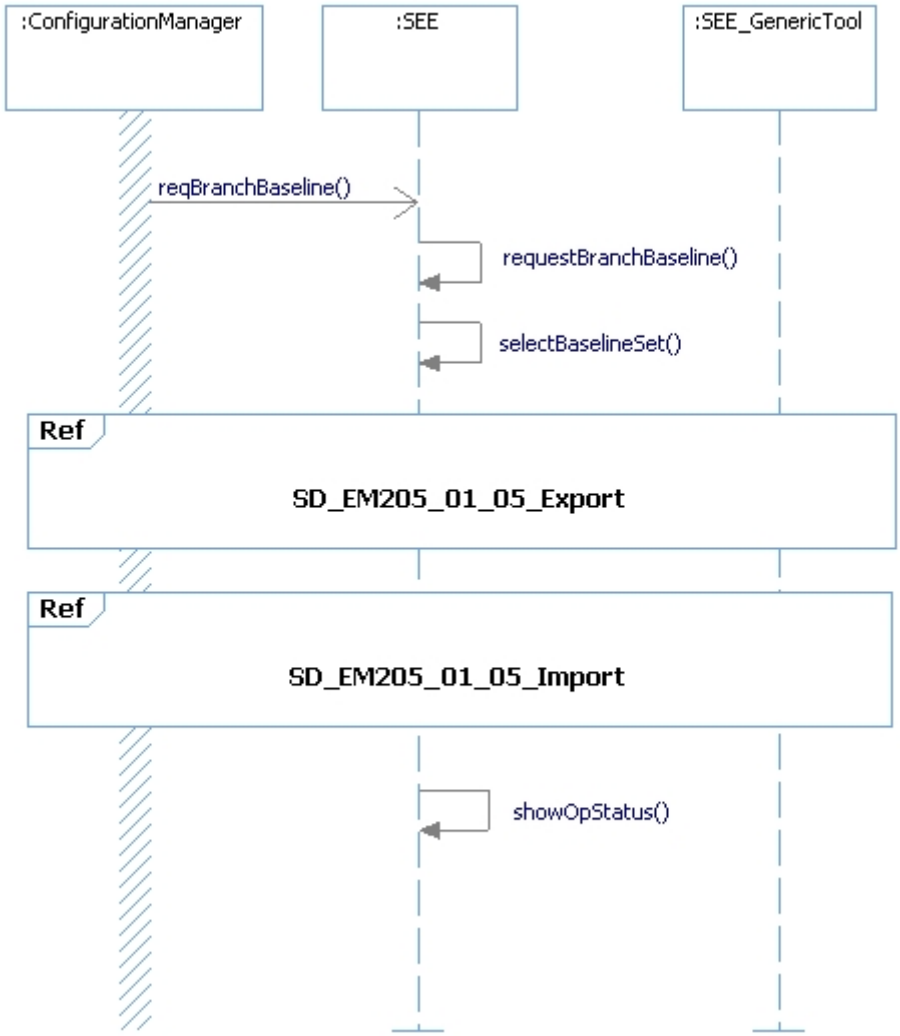

Figure 3-33 SD_EM205_01_02_BranchBaseline

requestBranchBaseline: The configuration manager requests to generate a branch from any baseline set.

Relates to Engineering Activity: 01. The configuration manager requests to generate a branch from any baseline set.

selectBaselineSet: The SEE prompts the user to select the baseline set containing the affected baseline of the configuration item.

Relates to Engineering Activity: 03. The SEE prompts the user to select a baseline set.

showOpStatus: Show the status of the complete operation.

Relates to Engineering Activity: 08. The SEE collects the operation status and displays the results.

3.4.1.4 EM205_01_03 - List Change History

ID: EM205_01_03

For given configuration items list the deltas and modifications from one baseline to the next baseline.

EM Purpose: To generate a delta history report.

EM Pre-Condition: Baseline set definitions are available and at least two baseline sets are established.

EM Post-Condition: A report is generated listing all the changes and modifications applied for the configuration items from one baseline set to a following baseline set.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The user requests to generate a history report.
02. At the SEE the user selects one baseline set from the available baseline sets. For each configuration item the SEE asks the tool if it supports the history feature.
03. The tool provides the information if it supports the history feature in general and for the current data base.
04. The SEE checks the history status.
05. If history is not supported, the SEE skips the configuration item and adds appropriate information in the report.
06. The SEE requests for the given configuration item the delta information to its previous baseline.
07. The tool provides the delta information for the given baseline of the configuration item.
08. The SEE adds the delta information to the report.
09. The user requests a full history and selects the baseline set definition. For each configuration item in all the baseline sets, the SEE asks the tool if it supports the history feature.
10. The tool provides the information if it supports the history feature in general and for the current data base.
11. The SEE checks the history status.
12. If history is not supported, the SEE skips the configuration item and adds appropriate information in the report.
13. The SEE requests for the given configuration item the delta information to its previous baseline.
14. The tool provides the delta information for the given baseline of the configuration item.
15. The SEE adds the delta information to the report.

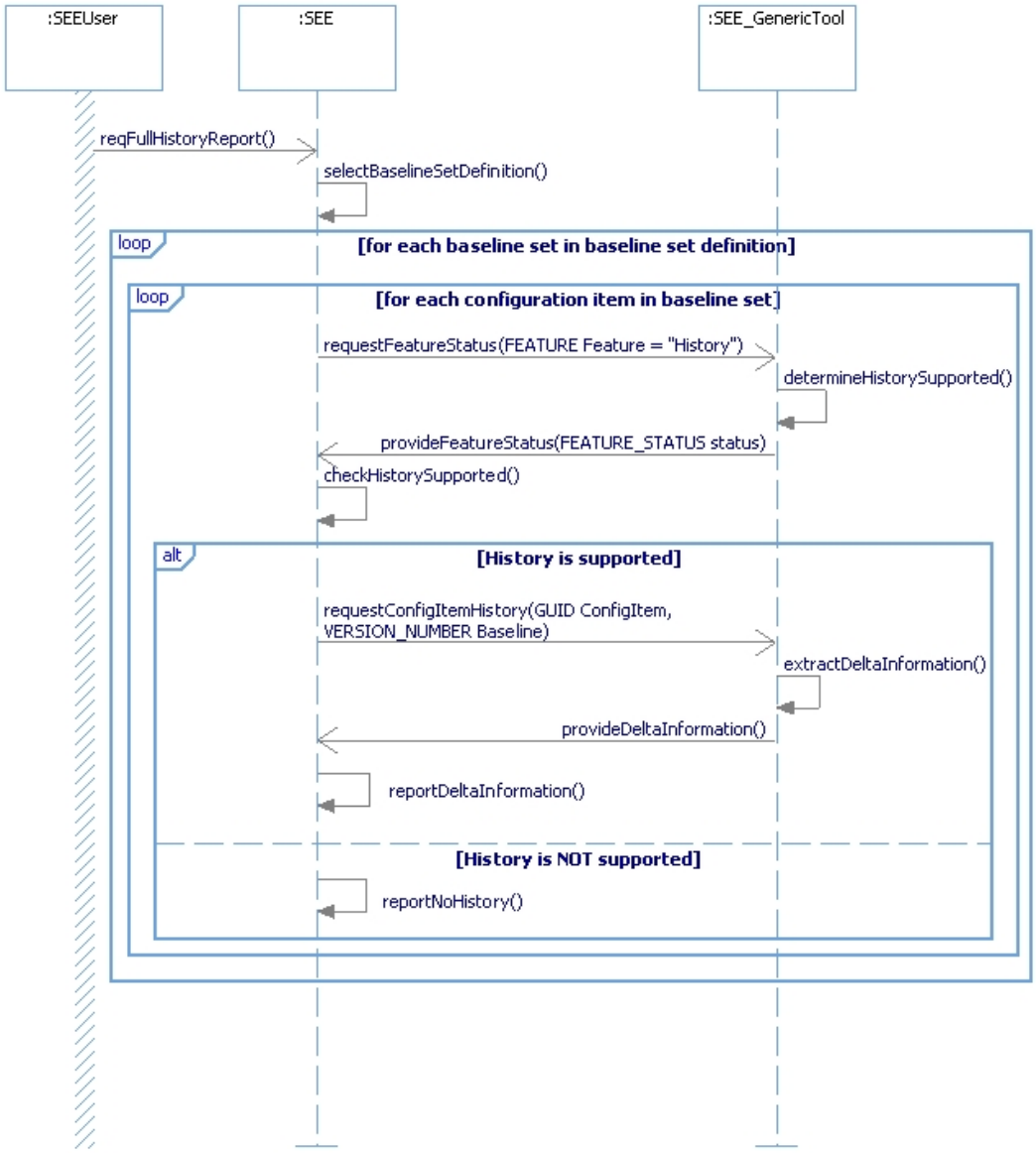

Figure 3-34 SD_EM205_01_03_FullHistoryReport

checkHistorySupported: The SEE checks if history feature is supported.

Relates to Engineering Activity: 11. The SEE checks the history status.

determineHistorySupported: The tool checks if the tool supports this feature and if maintenance for the current database is supported.

Relates to Engineering Activity: 10. The tool provides the information if it supports the history feature in general and for the current data base.

extractDeltaInformation: The tool extracts the delta information from the given baseline of the configuration item to its previous baseline.

Relates to Engineering Activity: 14. The tool provides the delta information for the given baseline of the configuration item.

reportDeltaInformation: The SEE adds the delta information to the report.

Relates to Engineering Activity: 15. The SEE adds the delta information to the report.

reportNoHistory: self-explanatory.

selectBaselineSetDefinition: The SEE prompts the user to select a baseline set definition containing the affected configuration item baseline.

Relates to Engineering Activity: 02. The SEE prompts the user to select a baseline set definition.

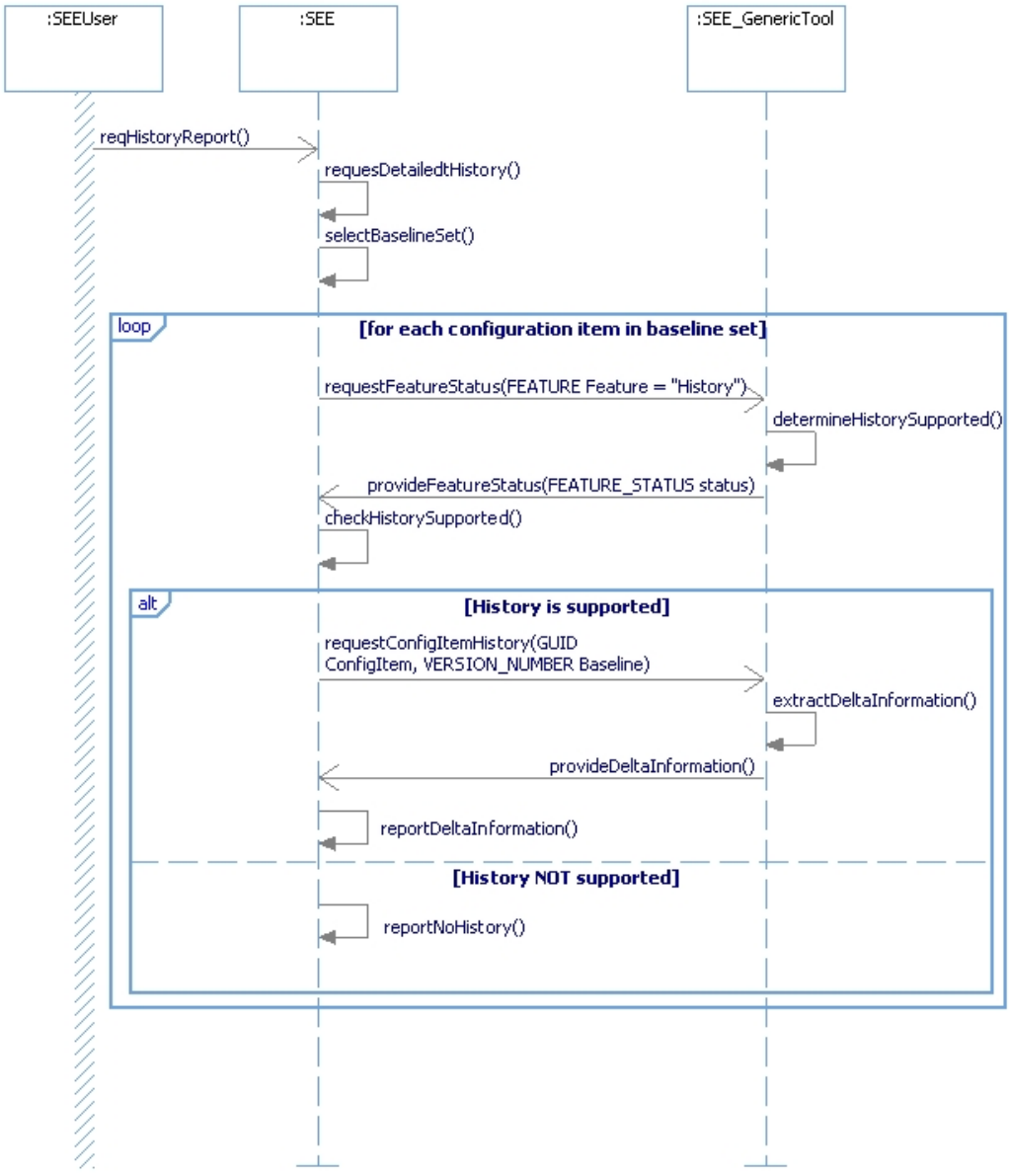

Figure 3-35 SD_EM205_01_03_HistoryReport

checkHistorySupported: The SEE checks if history feature is supported.

Relates to Engineering Activity: 11. The SEE checks the history status.

determineHistorySupported: The tool checks if the tool supports this feature and if maintenance for the current database is supported.

Relates to Engineering Activity: 10. The tool provides the information if it supports the history feature in general and for the current data base.

extractDeltaInformation: The tool extracts the delta information from the given baseline of the configuration item to its previous baseline.

Relates to Engineering Activity: 14. The tool provides the delta information for the given baseline of the configuration item.

reportDeltaInformation: The SEE adds the delta information to the report.

Relates to Engineering Activity: 15. The SEE adds the delta information to the report.

reportNoHistory: self-explanatory.

requesDetailedtHistory: self-explanatory.

selectBaselineSet: The SEE prompts the user to select the baseline set containing the affected baseline of the configuration item.

Relates to Engineering Activity: 03. The SEE prompts the user to select a baseline set.

3.4.1.5 EM205_01_04 - Merge Baselines

ID: EM205_01_04

Comparing different baseline sets of a baseline set definition is necessary to check which artefacts and configuration items have changed. For each artefact within a dedicated baseline set the version is listed and compared with its version in the other baseline set.
The user can select to open a viewer within the associated tool to show the differences.

EM Purpose: To integrate changes of different branches (work streams) into an other branch or the trunk.

EM Pre-Condition: Changes made to a branch

EM Post-Condition: Changes made in a branch (work stream) are integrated into the parent branch or the trunk.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The user requests the comparison of two baselines.
02. The user selects an available baseline set definition.
03. From the list of available baseline sets the user selects the two baseline sets to compare.
04. The SEE lists all the configuration items with the version number for both baseline sets highlighting different version numbers.
05. The configuration manager checks the changes made
06. The configuration manager has decided to replace a configuration item completely. The tool is requested to replace the configuration item.
07. The tool replaces the identified configuration item in the parent branch or trunk with the modified item from the branch. The tool provides an operation status,
08. The SEE collects the operation status and displays the results.
09. The user selects a configuration item and requests a detailed comparison.
11. The tool loads the two baselines of the given configuration item.
12. The tool visualizes the detailed changes of the two baselines.
13. The configuration manager accepts those changes to be incorporated from the branch into its parent branch or the trunk.
14. The tool incorporates the selected modification.
15. The user continues until all changes from the baseline are incorporated.
16. The configuration manager may decide to cancel the branch or to continue the branch. In case the branch is cancelled, working on that branch is no more possible.

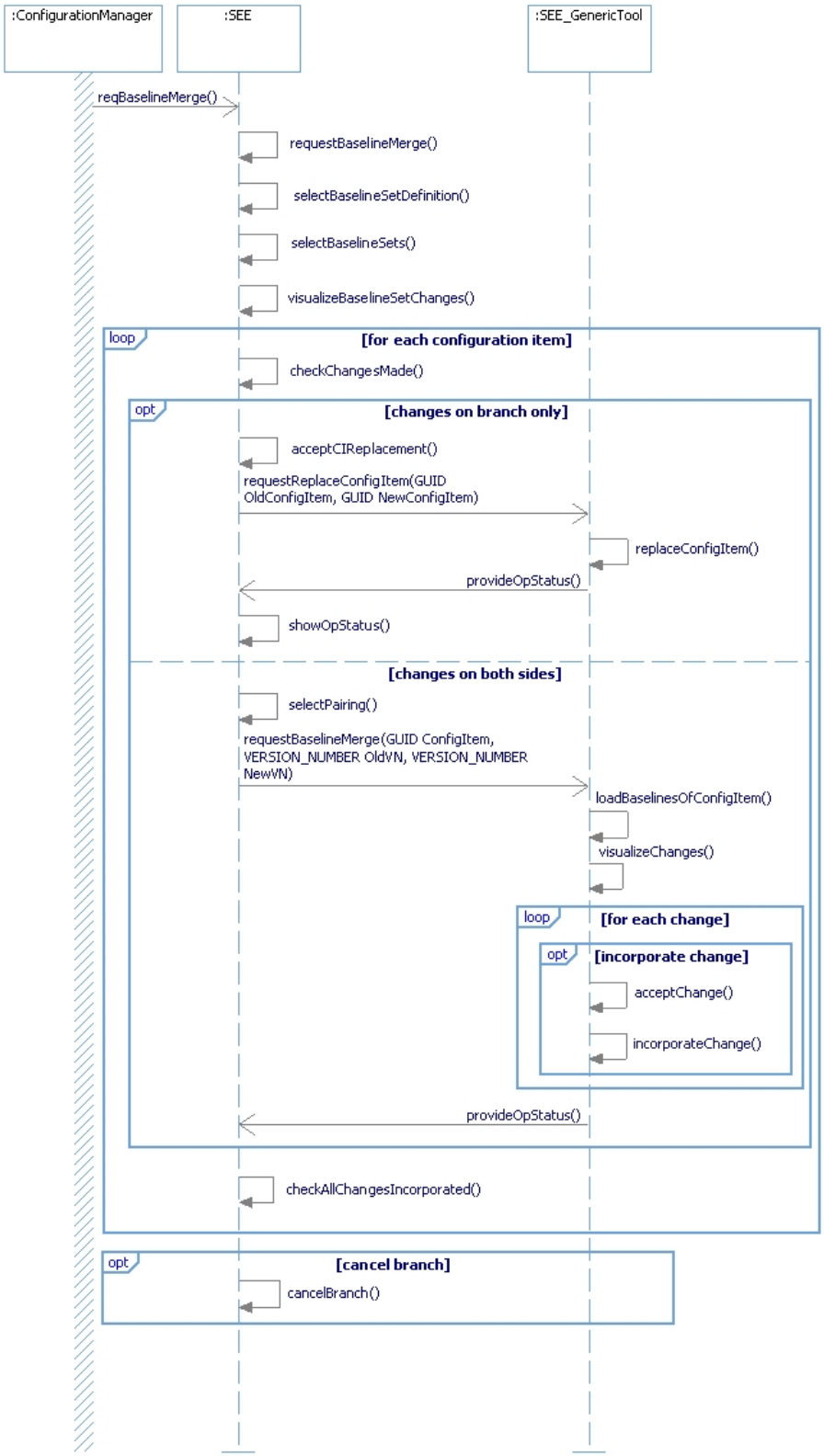

Figure 3-36 SD_EM205_01_04_MergeBaselines

acceptChange: The configuration manager decides for each modification, if it needs to be copied from the branch to the trunk or parent or if a modification is rejected.

Relates to Engineering Activity: 13. The configuration manager accepts those changes to be incorporated from the branch into its parent branch or the trunk.

acceptCIReplacement: Changes are found on the branch only. The configuration manager accepts that the configuration item from the branch replaces the configuration item on the trunk.

Relates to Engineering Activity: 06. The configuration manager has decided to replace a configuration item completely. The tool is requested to replace the configuration item.

cancelBranch: The configuration manager may decide to cancel the branch or to continue the branch. In case the branch is cancelled, working on that branch is not longer possible.

Relates to Engineering Activity: 16. The configuration manager may decide to cancel the branch or to continue the branch. In case the branch is cancelled, working on that branch is no more possible.

checkAllChangesIncorporated: The user continues until all changes from the baseline set in the selected branch are incorporated.

Relates to Engineering Activity: 15. The user continues until all changes from the baseline are incorporated.

checkChangesMade: The configuration manager checks the changes made. If changes are made on the branch only he may decide to replace the complete configuration item instead of integrating each change separately.

Relates to Engineering Activity: 05. The configuration manager checks the changes made

incorporateChange: The tool incorporates the selected modification.

Relates to Engineering Activity: 14. The tool incorporates the selected modification.

loadBaselinesOfConfigItem: The tool loads the two baselines of the given configuration item.

Relates to Engineering Activity: 11. The tool loads the two baselines of the given configuration item.

replaceConfigItem: The SEE tool replaces the configuration item in the trunk or parent with the one from the branch.

Relates to Engineering Activity: 07. The tool replaces the identified configuration item in the parent branch or trunk with the modified item from the branch. The tool provides an operation status,

requestBaselineMerge: The user requests to merge two baseline set definitions.

Relates to Engineering Activity: 01. The user requests the comparison of two baselines.

selectBaselineSetDefinition: The SEE prompts the user to select a baseline set definition containing the affected configuration item baseline.

Relates to Engineering Activity: 02. The SEE prompts the user to select a baseline set definition.

selectBaselineSets: The user selects two baseline sets of the baseline set definition.

Relates to Engineering Activity: 03. From the list of available baseline sets the user selects the two baseline sets to compare.

selectPairing: The SEEUser selects a configuration item and requests the detailed visualization of the deltas.

Relates to Engineering Activity: 09. The user selects a configuration item and requests a detailed comparison.

showOpStatus: Show the status of the complete operation.

Relates to Engineering Activity: 08. The SEE collects the operation status and displays the results.

visualizeBaselineSetChanges: The SEE lists the configuration items and their version number for both baseline sets. Changes are highlighted.

Relates to Engineering Activity: 04. The SEE lists all the configuration items with the version number for both baseline sets highlighting different version numbers.

visualizeChanges: The tool compares the two versions and visualizes the detailed differences. The tool provides the status of the operation.

Relates to Engineering Activity: 12. The tool visualizes the detailed changes of the two baselines.

3.4.1.6 EM205_01_05 - Export Import Baseline Set

ID: EM205_01_05

Load data from a given baseline set into an already available or empty workspace for purpose of:

- Rollback
- Branch
- Analysis
- Reporting
- ...

Export all the data and artefacts allocated to the given baseline set and baseline version. The exported data package shall contain all the data necessary to re-establish a functional workspace after re-import. Generic data exchange format shall be considered.
For a supplier a tailored baseline set is generated, while the tailoring is performed on configuration item level.

Import all the data allocated to the given baseline set and baseline version. The imported data package re-establishes the functional workspace.

EM Purpose: To create a copy of the SEE data in the local file system for archiving and restoring the workspace from an archive.

EM Pre-Condition: Project file and baseline sets are available.

EM Post-Condition: Workspace and project files restored from previously exported archive.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The SEEUser selects an available baseline set.
02. The SEE activates the baseline set. See "loadArtefact".
03. The CM starts export by selecting a baseline set.
04. The SEE creates a new archive.
05. The SEE stores relevant information from the baseline set in the archive.
06. For each configuration item the SEE requests the associated tool to export the baseline as defined by the baseline set.
07. The tool exports the requested baseline of the given configuration item into the archive.
07. The tool exports the requested baseline of the given configuration item into the archive.
08. The SEE collects and combines the results of each operation.
09. The CM starts the import of a baseline and selects the source archive.
10. The SEE loads the information on the baseline set.
11. The SEE requests each tool to create a new workspace.

12. Each tool creates a new workspace e.g. an empty database.
13. The SEE collects and combines the results of each operation.
14. For each config item in the baseline set, the SEE requests the associated tool to import the configuration item from the source archive.
15. Each tool imports the requested baseline of the given configuration item from the source archive.
16. The SEE collects and combines the results of each operation.

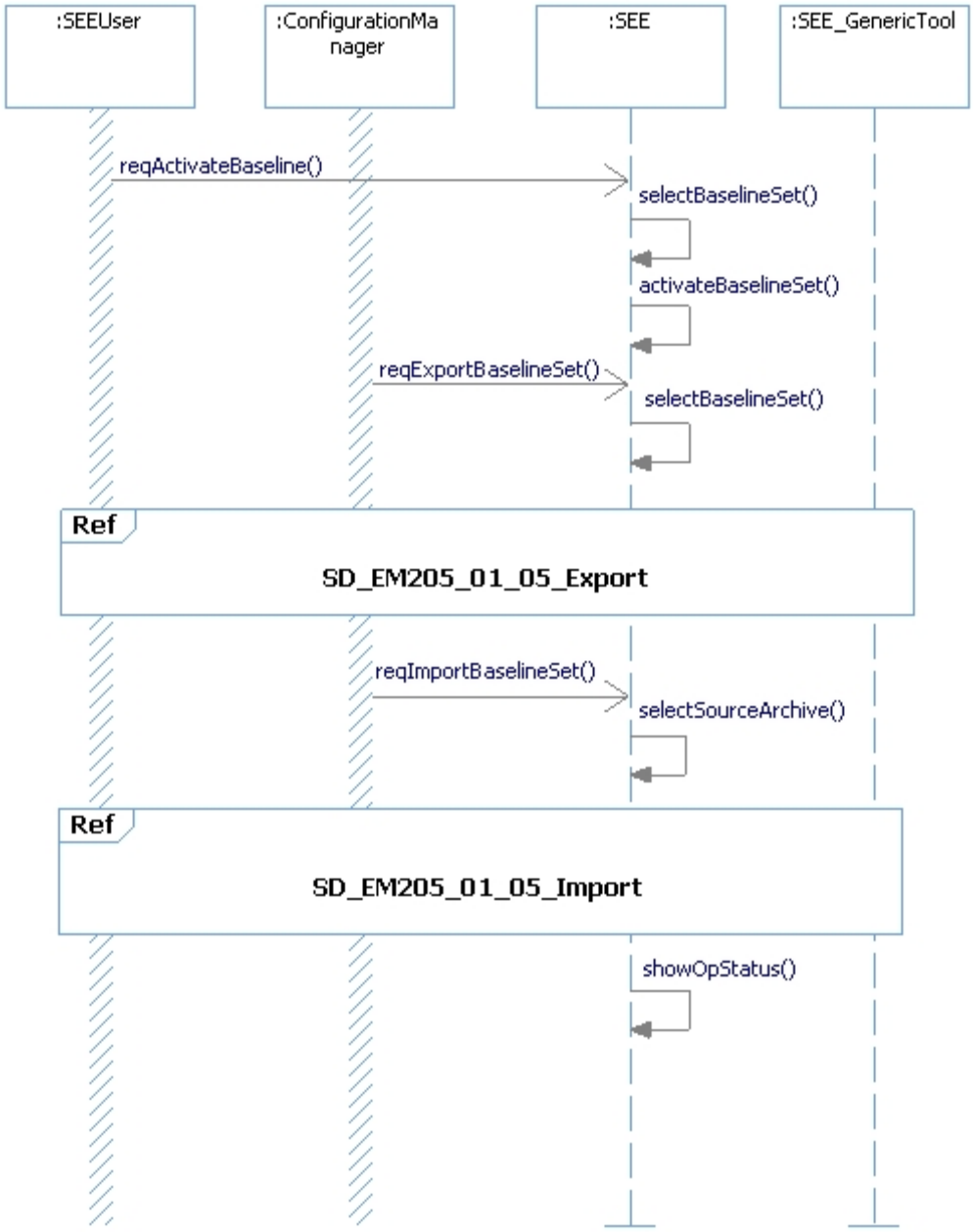

Figure 3-37 SD_EM205_01_05_TopExportImport

activateBaselineSet: The SEE activates the baseline set.

Relates to Engineering Activity: 02. The SEE activates the baseline set. See "loadArtefact".

selectBaselineSet: The SEE prompts the user to select the baseline set containing the affected baseline of the configuration item.

Relates to Engineering Activity: 03. The SEE prompts the user to select a baseline set.

selectSourceArchive: The configuration manager requests to restore a baseline from archive. The configuration manager selects the archive.

Relates to Engineering Activity: 09. The CM starts the import of a baseline and selects the source archive.

showOpStatus: Show the status of the complete operation.

Relates to Engineering Activity: 08. The SEE collects the operation status and displays the results.

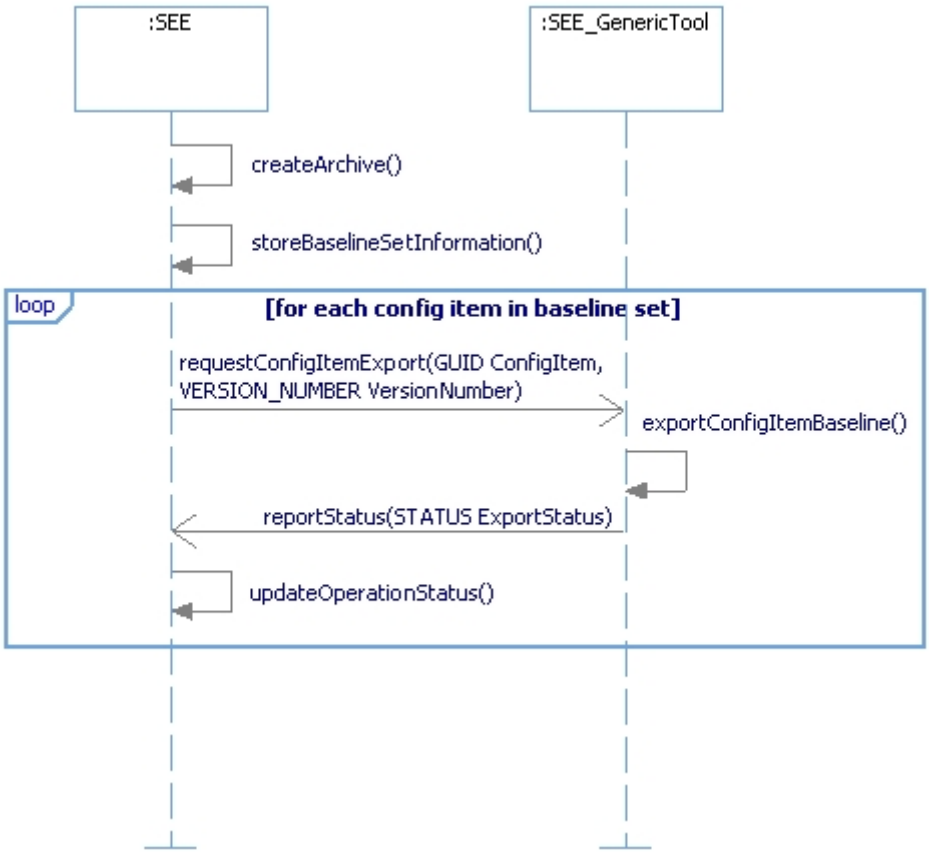

Figure 3-38 SD_EM205_01_05_Export

createArchive: The user selects a file system path and the SEE creates an archive structure.

Relates to Engineering Activity: 04. The SEE creates a new archive.

exportConfigItemBaseline: Each tool exports the given baseline of the configuration item to archive.

Relates to Engineering Activity: 07. The tool exports the requested baseline of the given configuration item into the archive.

storeBaselineSetInformation: The SEE stores information regarding the baseline set into the archive including the list of all configuration items with version number and associated information.

Relates to Engineering Activity: 05. The SEE stores relevant information from the baseline set in the archive.

updateOperationStatus: The SEE gathers the status of each operation and generates a summary status.

Relates to Engineering Activity: 16. The SEE collects and combines the results of each operation.

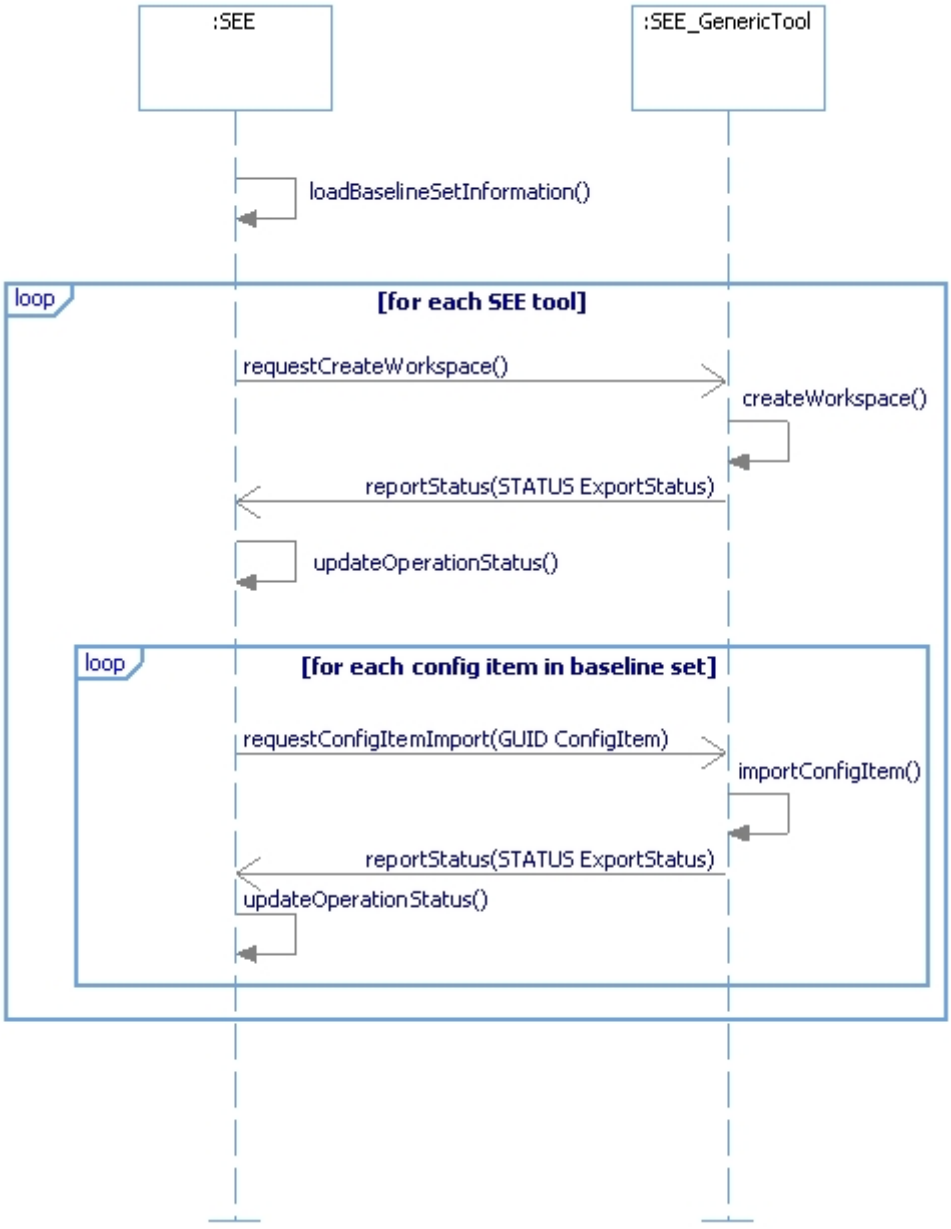

Figure 3-39 SD_EM205_01_05_Import

createWorkspace: Create in each tool a new workspace or database.

Relates to Engineering Activity: 12. Each tool creates a new workspace e.g. an empty database.

importConfigItem: The tool imports the given configuration item from the source archive.

Relates to Engineering Activity: 15. Each tool imports the requested baseline of the given configuration item from the source archive.

loadBaselineSetInformation: The SEE loads the baseline set information from the archive including the list of configuration items.

Relates to Engineering Activity: 10. The SEE loads the information on the baseline set.

updateOperationStatus: The SEE gathers the status of each operation and generates a summary status.

Relates to Engineering Activity: 16. The SEE collects and combines the results of each operation.

3.4.1.7 EM205_01_06 - Establish Baseline Sets

ID: EM205_01_06

Establish a baseline set by baselining all the configuration items listed in the baseline set definition. The baseline set assures the consistency of the baselined configuration items.

EM Purpose: Baseline all the configuration items included in the selected baseline set definition.

EM Pre-Condition: Configuration Items are available

EM Post-Condition: Configuration items as defined by the baseline set definition are baselined.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

05. Triggered by a a request to establish a baseline the SEE shows a list of all Baseline Set Definitions.

06. Select a baseline set definition and create an instance i.e. a Baseline Set and "open" it.

07a. Select a config item from the list of config items and request from CM tool the list of assigned and open CR.

07b. The CM tool provides a list of all open CR for the requested config item.

07c. The SEE checks open CR. If there are no CR open request the associated tool to create a baseline for config item.

07d. The tool baselines the requested config item and generates a new version number. The tool provides the new version number.

07e. The SEE store in the Baseline Set the version number for the config item.

08a. The systems engineer opens a config item in the dedicated tool and selects from an available baseline. The tool requests from the SEE to register this baseline in the open baseline set.

08b. The user selects an available baseline.

08c. The SEE stores the version number with the requested config item in the baseline set.

09. The systems engineer checks for completeness of the baseline set.

10. If all config items are baselined the Systems Engineer closes the baseline.

11. The systems engineer requests a list of all available baseline sets with associated information.

12. The systems engineer requests a list of all config item with associated information for a selected baseline set.

13. The systems engineer selects a dedicated config item and requests to open the baseline of it in the associated tool.

14. The associated tool opens the config item for read only.

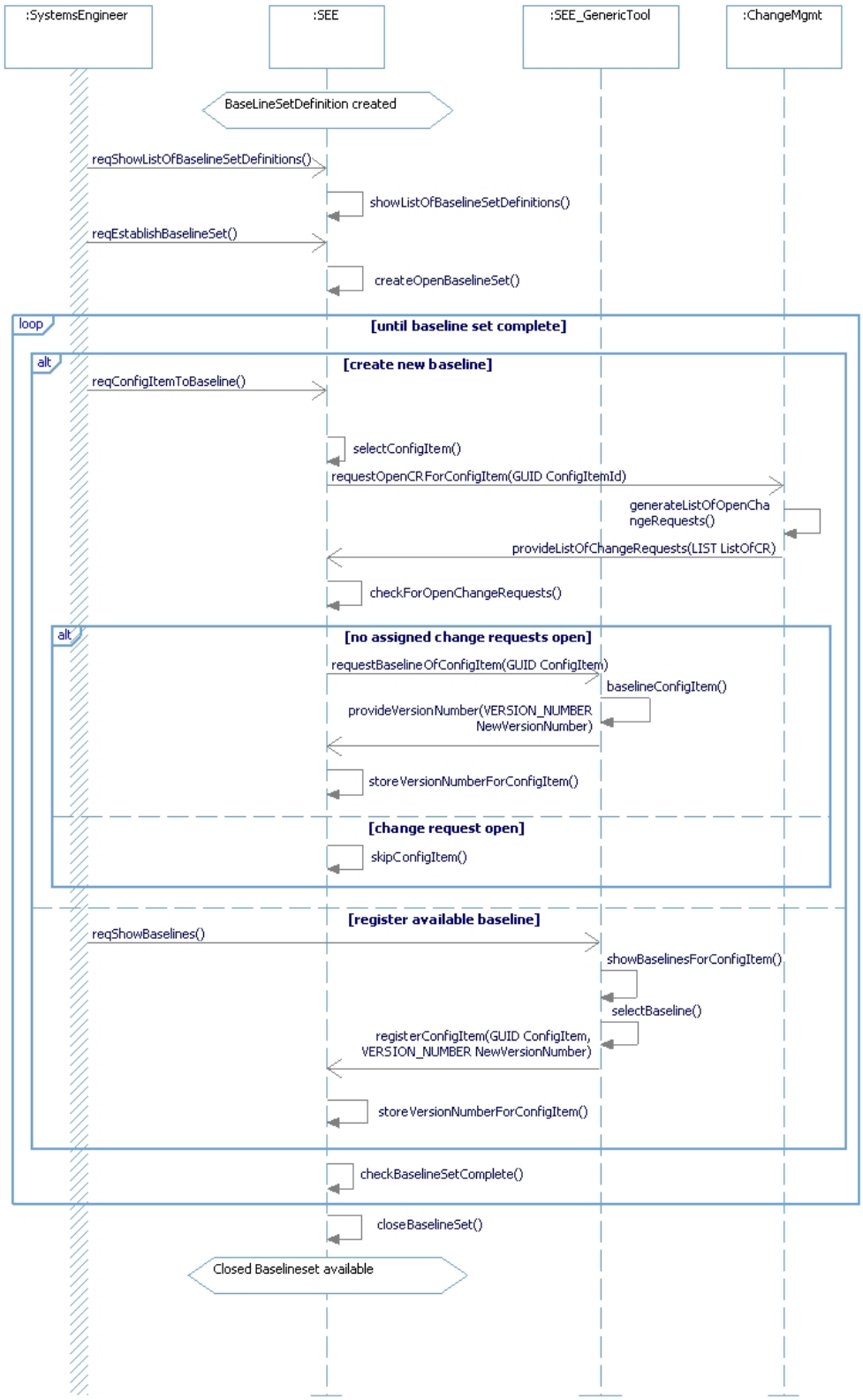

Figure 3-40 SD_EM205_01_06_EstablishBaseline

baselineConfigItem: Baseline a configuration item i.e. store in the database a copy of the current module and associated data. Increment the version number.

Relates to Engineering Activity: 07d. The tool baselines the requested config item and generates a new version number. The tool provides the new version number.

checkBaselineSetComplete: The process is repeated until all of the needed configuration items are baselined and the baseline set is complete.

Relates to Engineering Activity: 09. The systems engineer checks for completeness of the baseline set.

checkForOpenChangeRequests: The SEE checks open CR. If there are change requests assigned the SEE returns access to create an artefact for the parent configuration item. Otherwise, it returns no create access.

Relates to Engineering Activity: 11. The SEE checks open CR. If there are CR open the SEE returns access to create an artefact for the parent configuration item. Otherwise, it returns no create access.

closeBaselineSet: Close the baseline set. No more changes can be made to the baseline set i.e. no more configuration items can be included.

Relates to Engineering Activity: 10. If all config items are baselined the Systems Engineer closes the baseline.

createOpenBaselineSet: Create a baseline set based on the selected baseline set definition. The baseline set is "opened." Baselines of the dedicated configuration items may be added to the baseline set.

Relates to Engineering Activity: 06. Select a baseline set definition and create an instance i.e. a Baseline Set and "open" it.

generateListOfOpenChangeRequests: The change management extracts from the data base all open change requests assigned for the given configuration item. The change management provides a list of all change requests found.

Relates to Engineering Activity: 10. The CM tool provides a list of all open CR for the requested config item.

selectBaseline: The user selects a dedicated baseline from the list.

Relates to Engineering Activity: 08b. The user selects an available baseline.

selectConfigItem: The SEE prompts the user to select the affected configuration item from the baseline set. The SEE provides the selected configuration item.

Relates to Engineering Activity: 04. The SEE prompts the user to select a configuration item and provides the selected one.

showBaselinesForConfigItem: Show a list of all the baselines of a dedicated configuration item.

Relates to Engineering Activity: 08a. The systems engineer opens a config item in the dedicated tool and selects from an available baseline. The tool requests from the SEE to register this baseline in the open baseline set.

showListOfBaselineSetDefinitions: Show the contents of a baseline set definition list, i.e. listing all the configuration items hosted by the different tools and allocated to the given baseline set definition.

Relates to Engineering Activity: 05. Triggered by a a request to establish a baseline the SEE shows a list of all Baseline Set Definitions.

skipConfigItem: For analysis tools it is not necessary to support a history as on request they generate a complete report. The report is put under config control as a whole and usually there is no need to change the report manually authorized by change request.

Relates to Engineering Activity: 12. If history is not supported, the SEE skips the configuration item and adds an appropriate information in the report.

storeVersionNumberForConfigItem: Store in the baseline set the version numbers for the configuration item.

Relates to Engineering Activity: 08c. The SEE stores the version number with the requested config item in the baseline set.

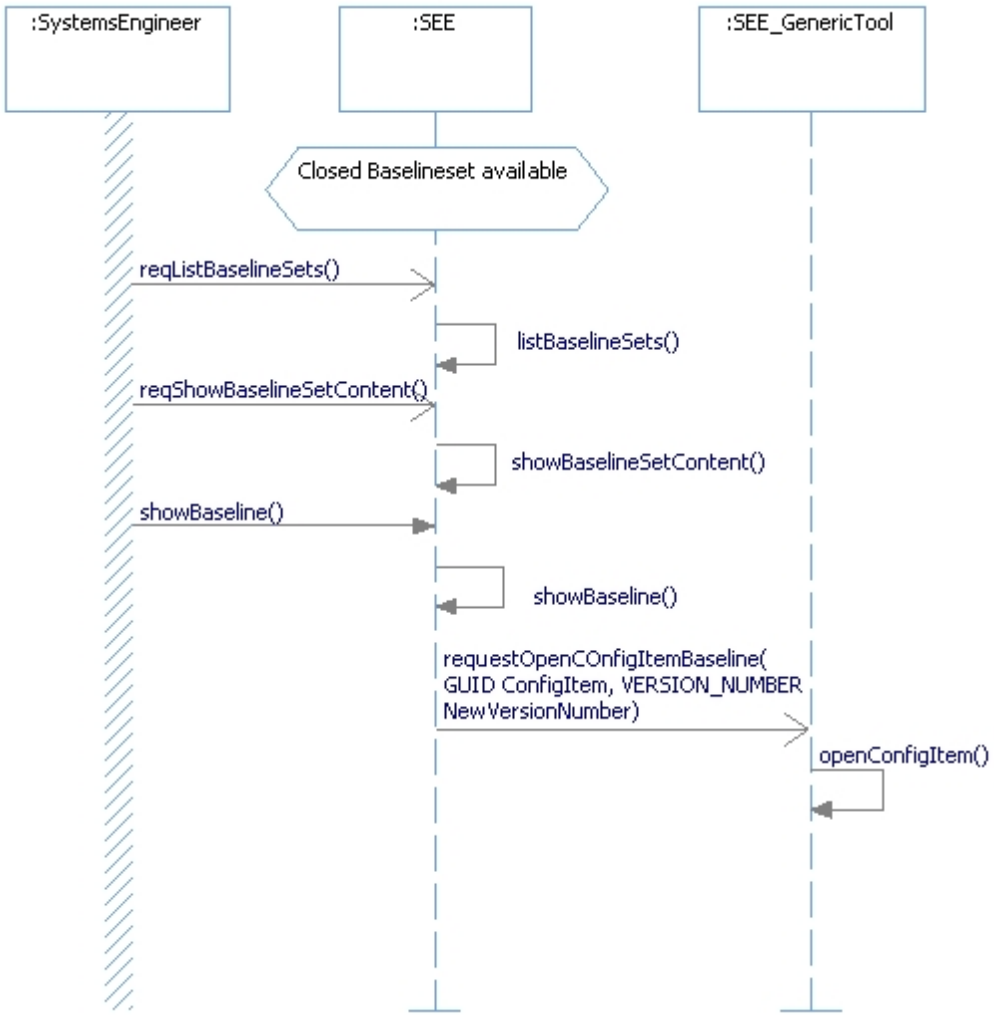

Figure 3-41 SD_EM205_01_06_ShowBaseline

listBaselineSets: List all the established baseline sets and their status, version number... for the selected baseline set definition.

Relates to Engineering Activity: 11. The systems engineer requests a list of all available baseline sets with associated information.

openConfigItem: The tool opens the requested configuration item with the given version number in read only mode.

Relates to Engineering Activity: 14. The associated tool opens the config item for read only.

showBaseline: Select a dedicated configuration item of the baseline set and open the identified baseline of the configuration item in the associated tool with read only access.

Relates to Engineering Activity: 13. The systems engineer selects a dedicated config item and requests to open the baseline of it in the associated tool.

showBaselineSetContent: List the baselines of all configuration items allocated to the given baseline set.

Relates to Engineering Activity: 12. The systems engineer requests a list of all config item with associated information for a selected baseline set.

3.4.1.8 EM205_01_07 - List Revision History

ID: EM205_01_07

List the revision history of an artefact showing the different versions and baselines, the assigned labels and therefore baselines and the change requests solved with each version.

EM Purpose: To list the revisions and change requests implemented for a dedicated baseline set.

EM Pre-Condition: Baseline set definitions are available and baseline sets are established.

EM Post-Condition: N/A

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The user requests to generate a revision history.
02. The user selects the baseline set from a list of all baseline sets of a dedicated baseline set definition.
03. The SEE creates the report and requests for each configuration item of the baseline set the relevant change requests.
04. The change management tool extracts all relevant change requests.
05. The SEE collects the change requests and requests further information for each.
06. The change management tool extracts and provides the information for the given change request.
07. The SEE updates the report with the received information.

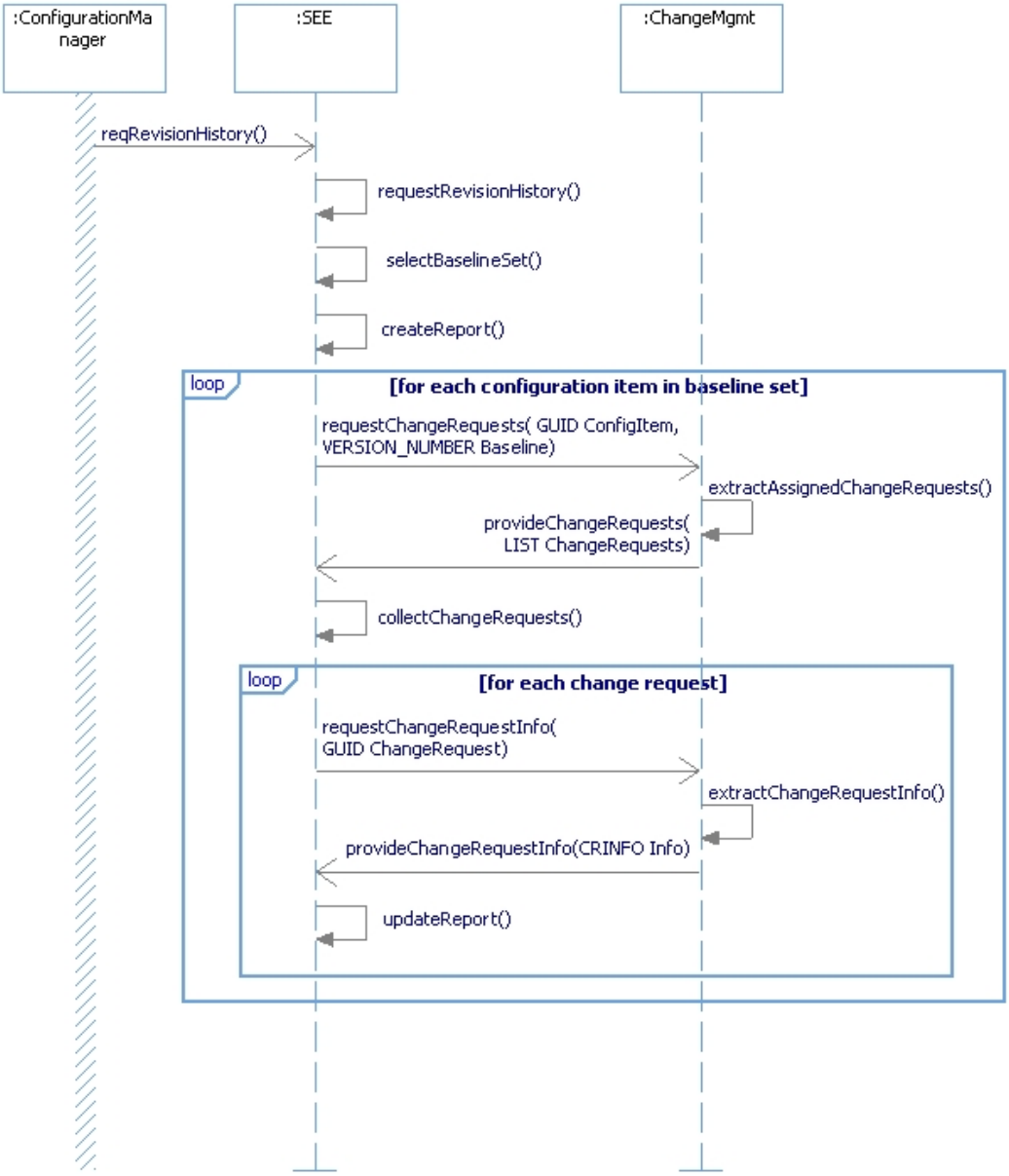

Figure 3-42 SD_EM205_01_07_ListRevisionHistory

collectChangeRequests: The SEE collects the change requests for all configuration items.

Relates to Engineering Activity: 05. The SEE collects the change requests and requests further information for each.

createReport: The SEE generates a report and requests each SEE tool to provide the version information.

Relates to Engineering Activity: 02. The SEE generates a report and requests each SEE tool to provide its version information

extractAssignedChangeRequests: The change management tool extracts all the change requests implemented, closed with the given baseline of the configuration item and the still open change requests for a baseline or the head of a working stream. The tool provides a list of all relevant change requests including their status (open, closed, ...).

Relates to Engineering Activity: 04. The change management tool extracts all relevant change requests.

extractChangeRequestInfo: The change management tool extracts and provides the information for the given change request.

Relates to Engineering Activity: 06. The change management tool extracts and provides the information for the given change request.

requestRevisionHistory: The user requests to generate a revision history for a dedicated baseline set or the current head of a branch or work stream.

Relates to Engineering Activity: 01. The user requests to generate a revision history.

selectBaselineSet: The SEE prompts the user to select the baseline set containing the affected baseline of the configuration item.

Relates to Engineering Activity: 03. The SEE prompts the user to select a baseline set.

updateReport: The SEE writes the received version information to the report.

Relates to Engineering Activity: 05. The SEE writes the received version information to the report.

3.4.2 PA205_02 - Change Control

ID: PA205_02

Related User Story: Process Automation, Guidance and Monitoring

Change control ensures that changes to a product or system are introduced in a controlled and coordinated manner. It reduces the possibility that unnecessary changes will be introduced to a system without forethought, introducing faults into the system or undoing changes made by other users of software.

It is assumed, that changes managed in the SEE only affect the artefacts within the SEE like models, requirements, or test cases. Changes to HW and SW are not managed here.

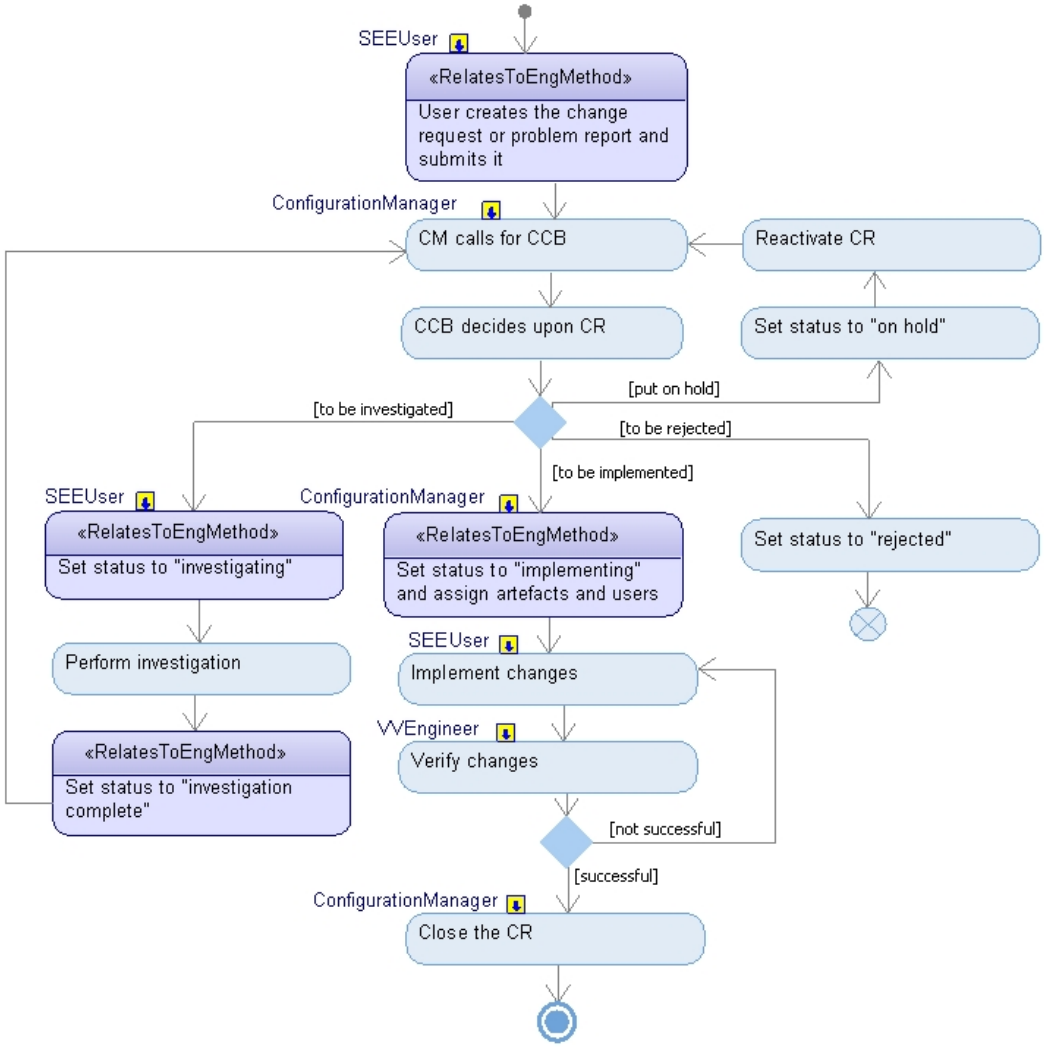

Figure 3-43 AD_PA205_02

CCB decides upon CR: The CCB decides upon the next steps with respect to each change request presented.

The CCB has the possibilities to set the change request to

"investigating" if the CCB has identified that additional investigation is necessary. This task is assigned to a dedicated team member.

"on hold" if the CCB has decided to postpone the resolution of the change request to a later revision.

"rejected" if the CCB has decided that the CR will never be solved, the identified problem is not a problem, the problem is allocated to an other system or the issue is covered by a different CR.

"implementing", if the CR is assigned to a dedicated team member for implementation of the changes.

Close the CR: The configuration manager closes the change request if the artefact is baselined. To close the issue or problem the customer or his internal representatives needs to approve the solution. Therefore, at least changes present in the customer products can only be closed if the change is incorporated in a product and release.

Remember, not every change is considered for the next release but for a subsequent release.

CM calls for CCB: The configuration manager is responsible of all the initiated change requests and calls in for the Configuration Control Board (CCB).

Implement changes: The commissioned user modifies the identified artefacts and implements the change solution. After the user has completed the implementation he sets the CR to "implemented". Verification personnel is informed.

Perform investigation: The purpose of the investigation is to identify the solution of the identified issue. It is not to fix it. The investigation shall also identify all the work items and artefacts that need to be changed directly. A change impact analysis is performed to identify all the work items and artefacts, which need to be updated and modified as a consequence of the change e.g. all the items linked to a changed requirement or analysis need to be updated due to changes in the design.

The goal is to identify clearly the extent of a change.

Reactivate CR: According to the recorded criteria the configuration manager reactivates the Change Request.

Set status to "implementing" and assign artefacts and users (RelatesToEngMethod): The configuration manager selects all the artefacts that need to be changed to implement the solution and assigns the change request to the configuration items. Only with a change request assigned the configuration items may be modified.

The changes request are set to implemented only if the change is required for the next baseline. Otherwise the change is not yet commissioned.

Based on the change impact analysis the configuration manager assigns first the artefacts, which need to be changed and modified. Then the configuration manager assigns the user who is commissioned to implement the changes.

In a strict change control environment only the assigned artefacts are allowed to be changed by the assigned user. For other users the artefacts are read only.

The assigned user is informed that he needs to implement the changes.

Set status to "investigating" (RelatesToEngMethod): The Configuration Manager sets the status of the CR to "investigating". He selects a user that should perform the analysis. The user is informed by email.

Set status to "investigation complete" (RelatesToEngMethod): After completion of the analysis the user set the status of the change request to "investigation complete".

Set status to "on hold": The configuration manager sets the status of the CR to "on hold". Additional information is provided, when the CR is reactivated.

Set status to "rejected": The configuration manager sets the status of the Change Request to "rejected". Additional information and reason for rejection is provided for purpose of traceability.

User creates the change request or problem report and submits it (RelatesToEngMethod): User creates the change request or problem report concerning a dedicated configuration item. The user selects the configuration item and the baseline where the issue is recognized.

The configuration item could be any type of artefact managed in the SEE.

The user submits the change request. I.e. the change request is now visible throughout the team. Control and responsibility changes are transferred to the configuration manager. The configuration manager receives an information e.g. via email.

Verify changes: Verification personnel is verifying that the changes are implemented correctly and that the changes are really solving the problem. After having completed the verification and the verification is OK, the verification personnel will set the CR to "solved".

The CR is set to "implementing" if the implementation is not ok.

3.4.2.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

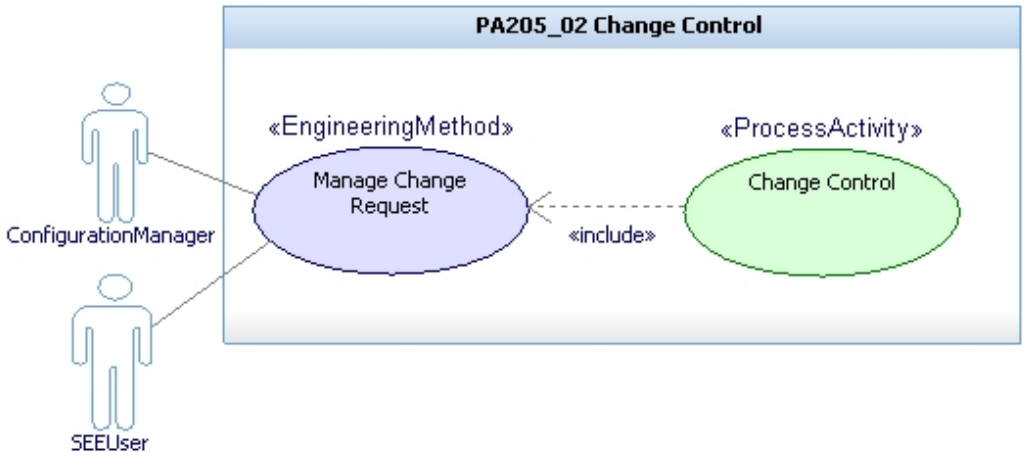

Figure 3-44 UCD_ChangeManagement

3.4.2.2 EM205_02_01 - Manage Change Request

ID: EM205_02_01

A change request may be generated by any SEEUser.

A change request has following different states:
"initiated" if the CR is created by a team member
"decision" if the CR need to be tackled in the next CCB

"investigating" the CCB has identified that additional investigation is necessary and the task is assigned to a dedicated team member

"on hold" the CCB has decided to postpone the resolution of the change request

"rejected" the CCB has decided that the CR will never be solved

"implementing" the CR is assigned to a dedicated team member for implementation of the changes

"verifying" the changes are verified to be correct

"solved" the changes are implemented as expected

"closed" the CR is closed with a new baseline

Whenever the status of a change request changes the relevant persons are informed by the system.

In the investigation phase a change impact analysis is performed to identify the artefacts that need to be changed.

If the change request is approved to be implemented, the state is set to implementing and the change request is assigned to the persons to implement the change and to the artefacts that need to be changed. The person is now allowed to open these artefacts and to modify them.

Assign the change request to a dedicated person and the artefacts affected.

In status "to be implemented"
The system drops a message to the implementing person. The artefacts allocated to the change request are unlocked for the assigned person in order to let that person change the artefacts. After the changes are complete, the change request is set to "implemented".

In status "implemented" a message is dropped to the person verifying the modifications. This persons checks the allocated artefacts for correct implementation of the change. If the modifications are ok the status of the change request is set to "Verified". The artefacts are locked again.

EM Purpose: To manage the implementation of changes to the data in a controlled manner.

EM Pre-Condition: Change control is activated for the project. Baseline Sets are available i.e. artefacts are baselined.

EM Post-Condition: Changes are incorporated into the database

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Create the Change Request and fill in relevant information.
02. The SEE prompts the user to select a baseline set definition.
03. The SEE prompts the user to select a baseline set.
04. The SEE prompts the user to select a configuration item and provides the selected one.
05. The change management completes the change request with a reference to the selected configuration item and baseline.
06. The change management requests the user with configuration control role from the UAM.
07. The UAM extracts the user with configuration management role from the database.
08. The change management requests to send an email to the configuration manager.
09. The emailer prepares the email and requests the email address for the given user from the UAM.
10. The UAM extracts the requested information form the database.
11. The emailer sends the email to the requested user and reports the status.

12. The change management reports the status.
13. The configuration manager assigns a task to a dedicated user. The change management tool requests a list of users from the UAM.
14. The UAM extracts a list of users and provides the list.
15. The change management tool prompts the configuration manager to select a user. The tool requests a email to send to the selected user.
16. The emailer prepares the email and requests the email address for the given user from the UAM.
17. The UAM extracts the requested information form the database.
18. The emailer sends the email to the requested user and reports the status.
19. The change management reports the status.
20. The user has completed the task and sets the status of the change request accordingly. The change management requests the user with configuration control role from the UAM.
21. The UAM extracts the user with configuration management role from the database.
22. The change management requests to send an email to the configuration manager.
23. The emailer prepares the email and requests the email address for the given user from the UAM.
24. The UAM extracts the requested information form the database.
25. The emailer sends the email to the requested user and reports the status.
26. The change management reports the status.
27. The configuration manager wants to commission the change to the dedicated user.
28. The change management tool requests a list of selected configuration items from the SEE.
29. The SEE prompts the user to select one of the available baseline set definitions.
30. The SEE provides the list of one or more selected configuration items.
31. For each configuration item the change management tool requests from the UAM a list of users having access to the configuration item.
32. The UAM extracts a list of users having access rights to the given configuration items. The UAM provides a list of valid users.
33. The change management tool prompts the user to select the user, who should be commissioned to change the configuration item.
34. The change management tool requests the UAM to set access rights for the selected user.
35. The UAM sets access rights to the given configuration item for the requested user.
36.
37. The emailer prepares the email and requests the email address for the given user from the UAM.
38. The UAM extracts the requested information form the database.
39. The emailer sends the email to the requested user and reports the status.
40. The change management reports the status.

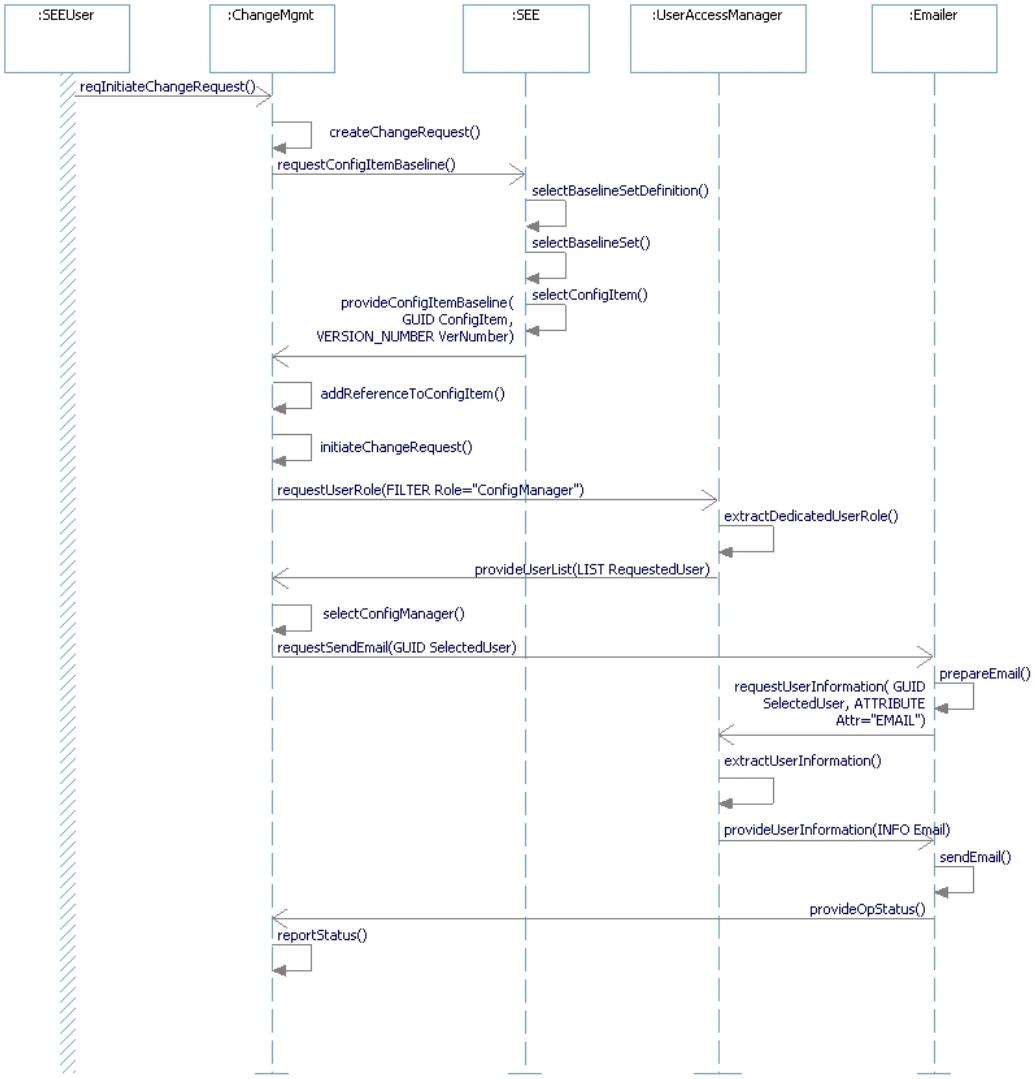

Figure 3-45 SD_EM205_02_01_InitiateChangeRequest

addReferenceToConfigItem: The CM tool adds to the change request the reference to the affected baseline of the configuration item. It is always a baseline to be selected. Changes to work items are not traced.

Relates to Engineering Activity: 05. The change management completes the change request with a reference to the selected configuration item and baseline.

createChangeRequest: A SEEUser creates a change request in the configuration control tool and adds a reference to the affected baseline of the configuration item.

Relates to Engineering Activity: 01. Create the Change Request and fill in relevant information.

extractDedicatedUserRole: The UAM extracts all the users from the database having the requested role. The UAM provides a list of users.

Relates to Engineering Activity: 07. The UAM extracts the user with configuration management role from the database.

extractUserInformation: The UAM extracts from the database and provides the requested user related information.

Relates to Engineering Activity: 38. The UAM extracts the requested information form the database.

initiateChangeRequest: The user has completed the data entry and initiates the change request, The change requests gets visible to the configuration manager. The configuration manger gets informed by email. The change request tool requests the configuration manager role from the UAM.

Relates to Engineering Activity: 06. The change management requests the user with configuration control role from the UAM.

prepareEmail: The emailer prepares the email and requests the email address of the requested addressee from the UAM.

Relates to Engineering Activity: 37. The emailer prepares the email and requests the email address for the given user from the UAM.

reportStatus: Show operation status to user.

Relates to Engineering Activity: 40. The change management reports the status.

selectBaselineSet: The SEE prompts the user to select the baseline set containing the affected baseline of the configuration item.

Relates to Engineering Activity: 03. The SEE prompts the user to select a baseline set.

selectBaselineSetDefinition: The SEE prompts the user to select a baseline set definition containing the affected configuration item baseline.

Relates to Engineering Activity: 02. The SEE prompts the user to select a baseline set definition.

selectConfigItem: The SEE prompts the user to select the affected configuration item from the baseline set. The SEE provides the selected configuration item.

Relates to Engineering Activity: 04. The SEE prompts the user to select a configuration item and provides the selected one.

selectConfigManager: The change management tool prompts the user to select a configuration manager from the list provided. Usually there is only one configuration manager identified for a database. In this case the list will contain only a single entry, which is automatically selected. The change management tool requests to send an email to the configuration manager.

Relates to Engineering Activity: 08. The change management requests to send an email to the configuration manager.

sendEmail: The emailer sends the email to the requested user and provides the operation status to the caller.

Relates to Engineering Activity: 39. The emailer sends the email to the requested user and reports the status.

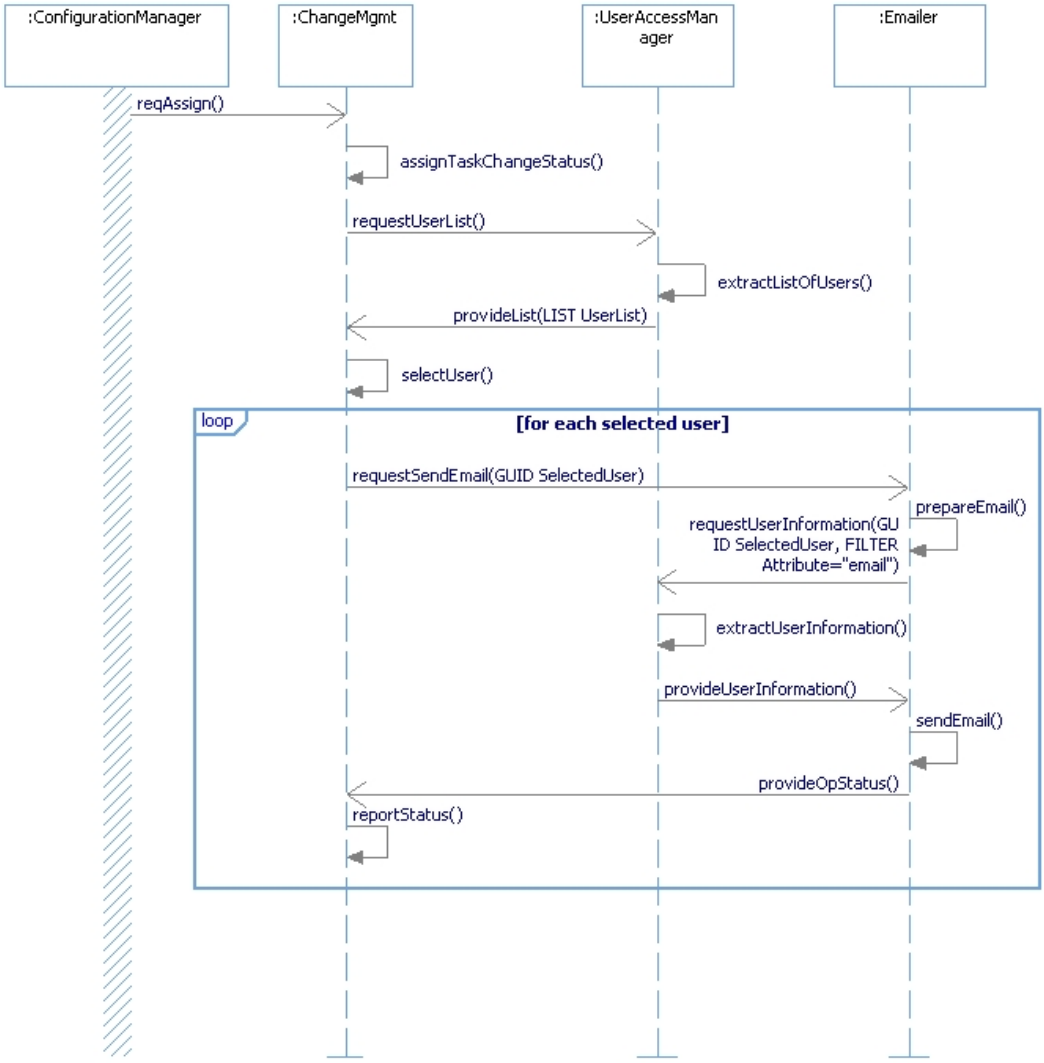

Figure 3-46 SD_EM205_02_01_AssignTask

assignTaskChangeStatus: The configuration manager commissions the investigation task to a dedicated user. The configuration management tool requests a list of users from the UAM.

Relates to Engineering Activity: 13. The configuration manager assigns a task to a dedicated user. The change management tool requests a list of users from the UAM.

extractListOfUsers: The UAM generates and provides a list of users allowed to change the artefact.

Relates to Engineering Activity: 32. The UAM extracts a list of users having access rights to the given configuration items. The UAM provides a list of valid users.

extractUserInformation: The UAM extracts from the database and provides the requested user related information.

Relates to Engineering Activity: 38. The UAM extracts the requested information form the database.

prepareEmail: The emailer prepares the email and requests the email address of the requested addressee from the UAM.

Relates to Engineering Activity: 37. The emailer prepares the email and requests the email address for the given user from the UAM.

reportStatus: Show operation status to user.

Relates to Engineering Activity: 40. The change management reports the status.

selectUser: For each configuration item, the change management tool prompts the user to select the user who shall implement the change.

Relates to Engineering Activity: 33. The change management tool prompts the user to select the user, who should be commissioned to change the configuration item.

sendEmail: The emailer sends the email to the requested user and provides the operation status to the caller.

Relates to Engineering Activity: 39. The emailer sends the email to the requested user and reports the status.

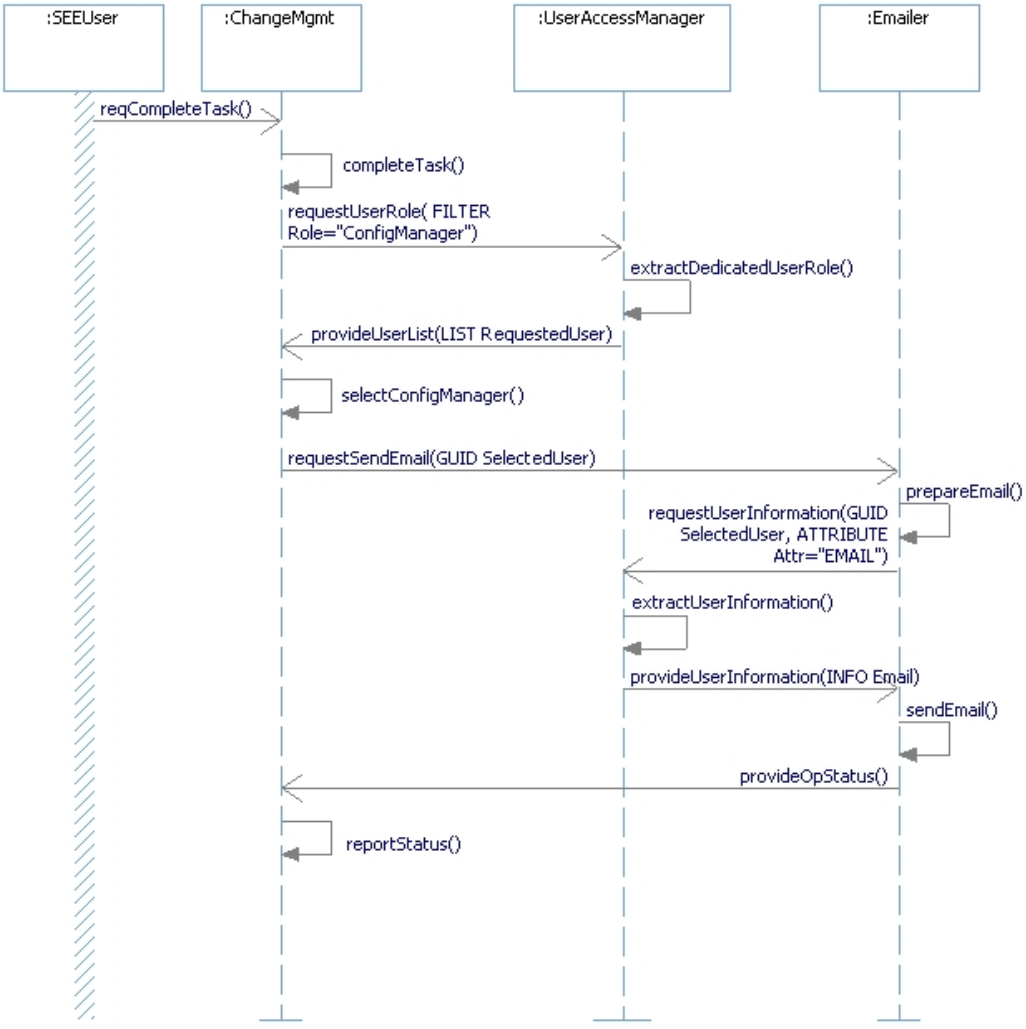

Figure 3-47 SD_EM205_02_01_CompleteTask

completeTask: The user has completed the commissioned task and set the status of the change request accordingly. The change requests again is allocated to the configuration manager. The configuration manger gets informed by email. The change request tool requests the configuration manager role from the UAM.

Relates to Engineering Activity: 20. The user has completed the task and sets the status of the change request accordingly. The change management requests the user with configuration control role from the UAM.

extractDedicatedUserRole: The UAM extracts all the users from the database having the requested role. The UAM provides a list of users.

Relates to Engineering Activity: 07. The UAM extracts the user with configuration management role from the database.

extractUserInformation: The UAM extracts from the database and provides the requested user related information.

Relates to Engineering Activity: 38. The UAM extracts the requested information form the database.

prepareEmail: The emailer prepares the email and requests the email address of the requested addressee from the UAM.

Relates to Engineering Activity: 37. The emailer prepares the email and requests the email address for the given user from the UAM.

reportStatus: Show operation status to user.

Relates to Engineering Activity: 40. The change management reports the status.

selectConfigManager: The change management tool prompts the user to select a configuration manager from the list provided. Usually there is only one configuration manager identified for a database. In this case the list will contain only a single entry, which is automatically selected. The change management tool requests to send an email to the configuration manager.

Relates to Engineering Activity: 08. The change management requests to send an email to the configuration manager.

sendEmail: The emailer sends the email to the requested user and provides the operation status to the caller.

Relates to Engineering Activity: 39. The emailer sends the email to the requested user and reports the status.

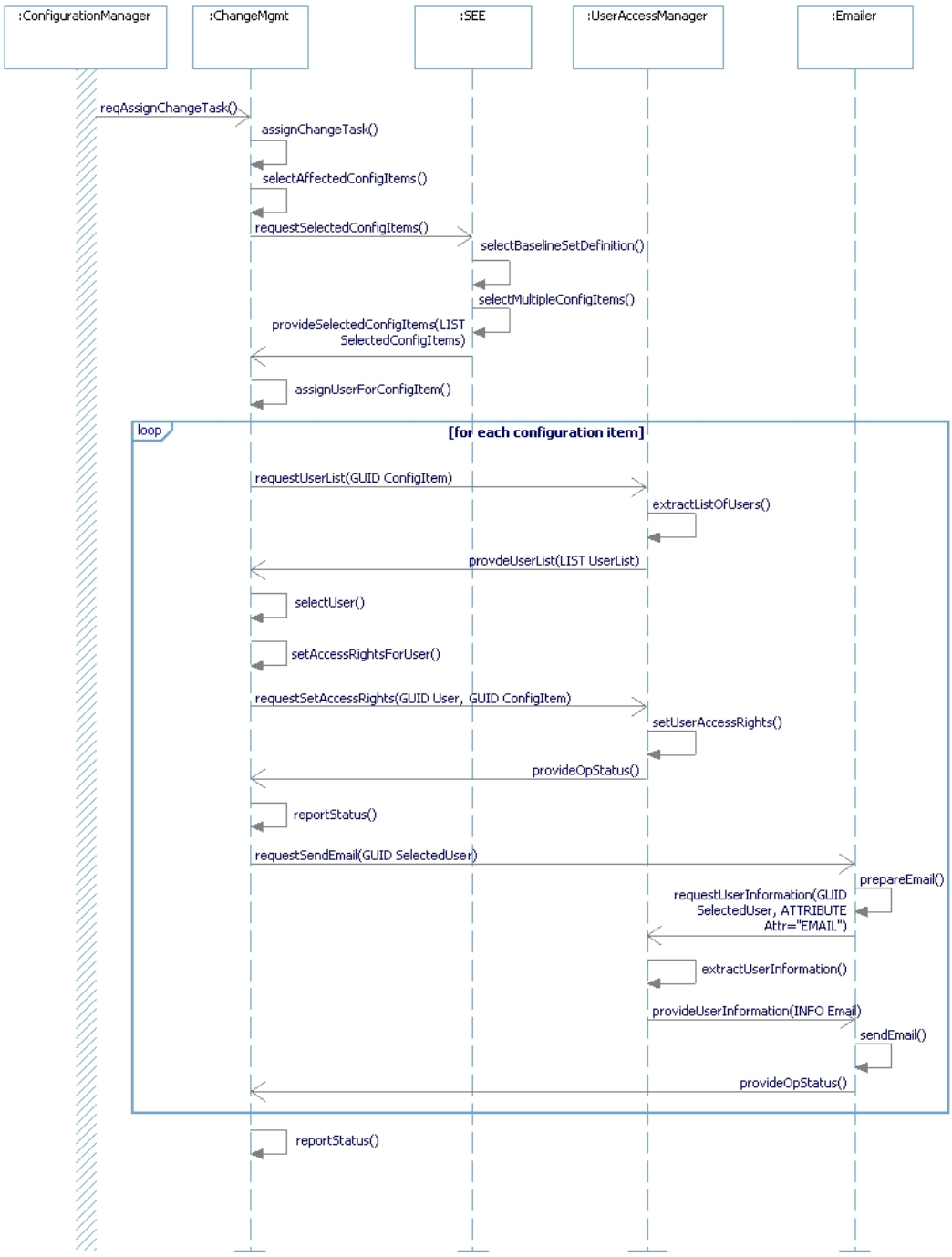

Figure 3-48 SD_EM205_02_01_AssignChangeTask

assignChangeTask: The identified changes shall be incorporated. The configuration manager assigns the task to change the artefacts to a subset of the SEEUsers considering individual access rights to the affected artefacts.

Relates to Engineering Activity: 27. The configuration manager wants to commission the change to the dedicated user.

assignUserForConfigItem: For each configuration item, the change management tool requests from the UAM a list of users having access rights for that configuration item.

Relates to Engineering Activity: 31. For each configuration item the change management tool requests from the UAM a list of users having access to the configuration item.

extractListOfUsers: The UAM generates and provides a list of users allowed to change the artefact.

Relates to Engineering Activity: 32. The UAM extracts a list of users having access rights to the given configuration items. The UAM provides a list of valid users.

extractUserInformation: The UAM extracts from the database and provides the requested user related information.

Relates to Engineering Activity: 38. The UAM extracts the requested information form the database.

prepareEmail: The emailer prepares the email and requests the email address of the requested addressee from the UAM.

Relates to Engineering Activity: 37. The emailer prepares the email and requests the email address for the given user from the UAM.

reportStatus: Show operation status to user.

Relates to Engineering Activity: 40. The change management reports the status.

selectAffectedConfigItems: The configuration manager selects the configuration items to be changed. The change management tool requests a list of affected configuration items from the SEE.

Relates to Engineering Activity: 28. The change management tool requests a list of selected configuration items from the SEE.

selectBaselineSetDefinition: The SEE prompts the user to select a baseline set definition containing the affected configuration item baseline.

Relates to Engineering Activity: 02. The SEE prompts the user to select a baseline set definition.

selectMultipleConfigItems: The SEE prompts the user to select from the baseline set definition one or more configuration items affected by the change. The SEE provides a list of all the selected configuration items.

Relates to Engineering Activity: 30. The SEE provides the list of one or more selected configuration items.

selectUser: For each configuration item, the change management tool prompts the user to select the user who shall implement the change.

Relates to Engineering Activity: 33. The change management tool prompts the user to select the user, who should be commissioned to change the configuration item.

sendEmail: The emailer sends the email to the requested user and provides the operation status to the caller.

Relates to Engineering Activity: 39. The emailer sends the email to the requested user and reports the status.

setAccessRightsForUser: For each configuration item, the change management tool requests to set in the UAM the access rights for that user to read / write allowing that user to change the configuration item.

Relates to Engineering Activity: 34. The change management tool requests the UAM to set access rights for the selected user.

setUserAccessRights: The UAM sets the access rights of the given user for the requested configuration to read / write access. That user is allowed to change the configuration item. In a strict controlled environment only one user at a time has the possibility to change a configuration item.

Relates to Engineering Activity: 35. The UAM sets access rights to the given configuration item for the requested user.

3.4.3 PA205_03 - User and User Access Management

ID: PA205_03

Related User Story: Process Automation, Guidance and Monitoring

User and User Access management deals with

- Authentication - verifying users are who they claim to be
- Authorization - granting users access to resources (also called entitlements)
- Auditing - recording who did what and when
- Administration - managing users and entitlements
- Confidentiality - protecting data from unauthorized eyes
- Notification - actively communicating security events

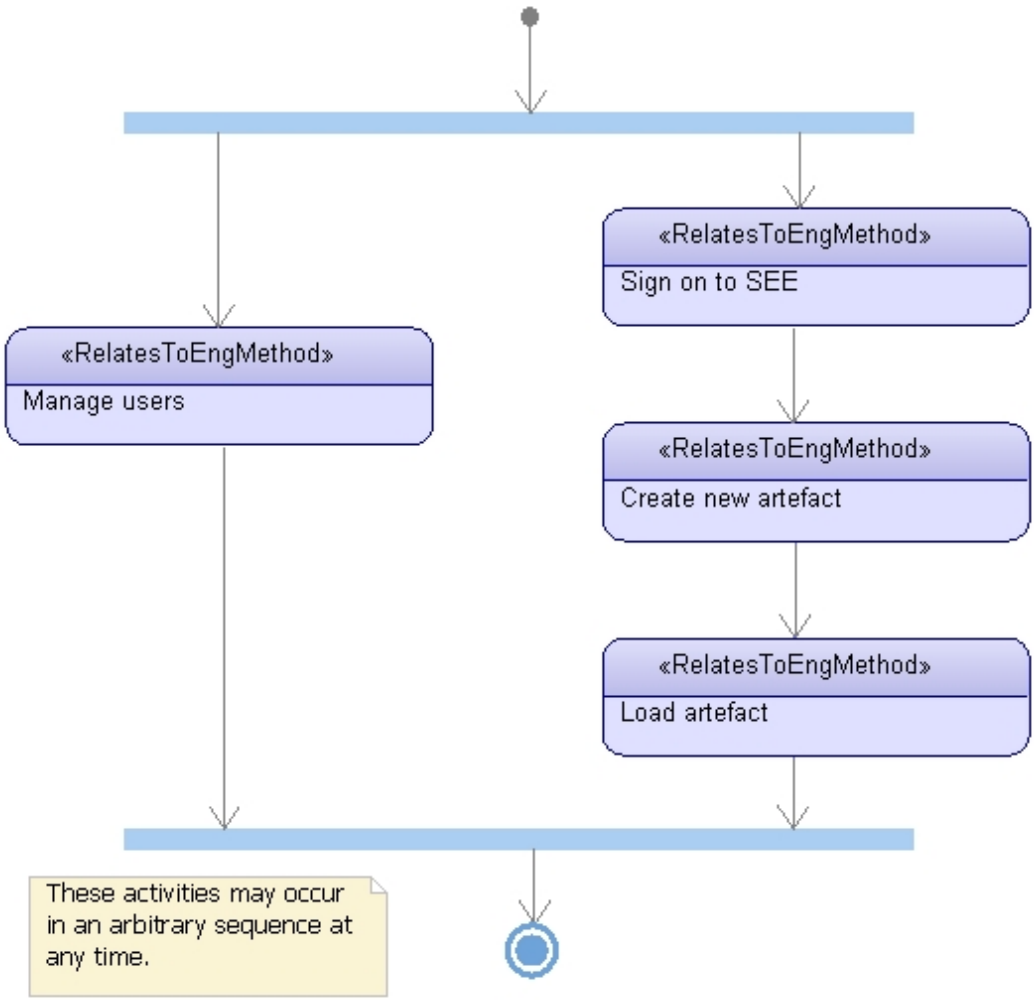

Figure 3-49 AD_PA205_03

Create new artefact (RelatesToEngMethod): The user signs on to the SEE. The UAM checks the access rights. This is a single sign on. One place, the user can sign on to all the integrated tools and databases. Once the user is signed on and he has appropriate rights he is able to create new artefacts in the database.

Load artefact (RelatesToEngMethod): Once signed on, the user is able to access and load other artefacts and elements from the SEE databases. The UAM checks the user access rights to the requested artefacts.

Manage users (RelatesToEngMethod): Manage user groups, users and the access rights to the different configuration items within the SEE. For each configuration item the access rights may be specified.

Before a user can access the database or create new artefact, the user need to be added to the User Access Management (UAM) database and access rights need to be assigned.

Sign on to SEE (RelatesToEngMethod): The user signs on to the SEE. The UAM checks the access rights. This is a single sign on. One place, the user can sign on to all the integrated tools and databases. Once the user is signed on and he has appropriate rights he is able to create new artefacts in the database.

3.4.3.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

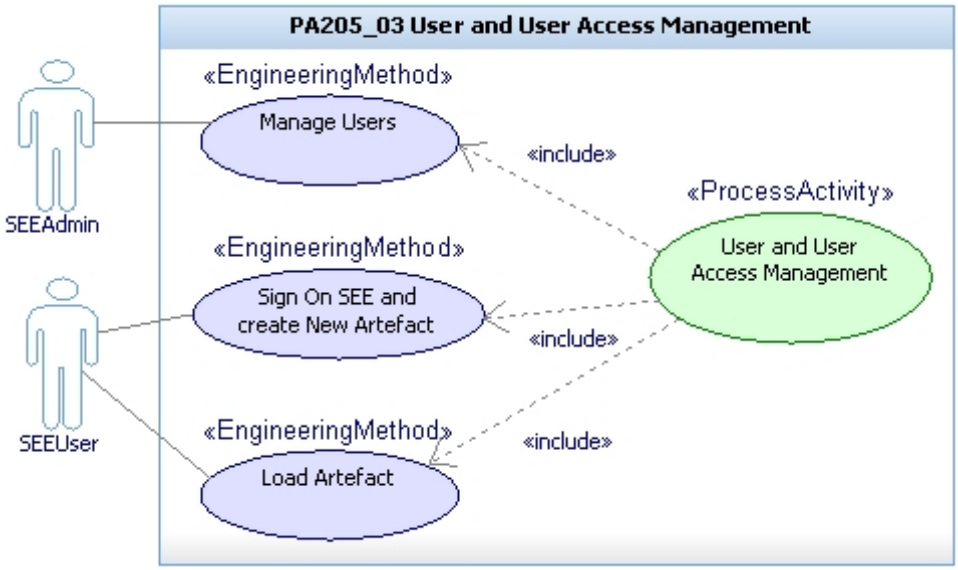

Figure 3-50 UCD_UserRights

3.4.3.2 EM205_03_01 - Manage Users

ID: EM205_03_01

Manage user groups, users and the access rights to the different configuration items within the SEE. For each configuration item the access rights may be specified.

EM Purpose: The engineering method covers the different activities to manage the users and user groups for the SEE.

User groups may be created, deleted and for each CI the access can be defined.

Users may be created, deleted and allocated to a defined user group.

EM Pre-Condition: N/A

EM Post-Condition: User groups and users are defined, known and permissions are allocated.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The SEE Administrator creates a user group within the UAM and requests a list of all SEE tools.
02. The SEE provides a list of tools known to the SEE.
03. The UAM requests from each tool a list of available configuration items.
04. The tool provides a list of all available configuration items.
05. The SEE Admin selects for each config item the access rights for the group.
06. The UAM stores the access rights information in the database.
07. The SEE Administrator creates a new user.
08. The SEE Admin allocates the new user to a pre-defined user group.
09. The SEE Administrator removes a user from the database.
10. The SEE Administrator user group from the data base.
11. The SEE Administrator changes the access rights for a dedicated config item already allocated to a user group.

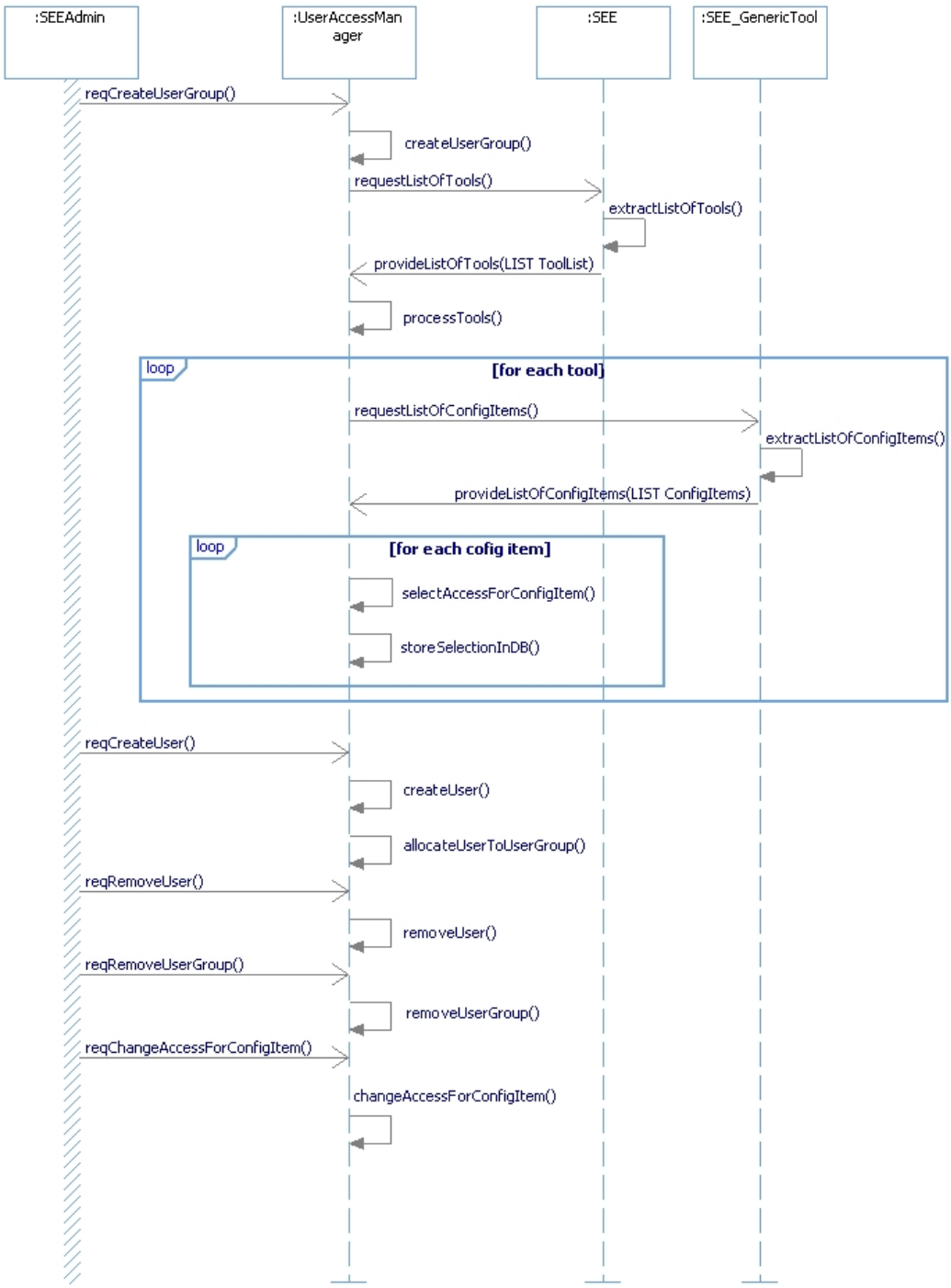

Figure 3-51 SD_EM205_03_01_ManageUserAccess

allocateUserToUserGroup: The SEE Administrator allocates the user to a pre-defined user group.

Relates to Engineering Activity: 08. The SEE Admin allocates the new user to a pre-defined user group.

changeAccessForConfigItem: The SEE Administrator changes the access rights for a dedicated config item already allocated to a user group.

Relates to Engineering Activity: 11. The SEE Administrator changes the access rights for a dedicated config item already allocated to a user group.

createUser: The SEE Administrator creates a new user.

Relates to Engineering Activity: 07. The SEE Administrator creates a new user.

createUserGroup: The SEE Administrator creates a user group within the UAM.

Relates to Engineering Activity: 01. The SEE Administrator creates a user group within the UAM and requests a list of all SEE tools.

extractListOfConfigItems: Extracts a complete set of configuration items included in the currently loaded database.

Relates to Engineering Activity: 04. The tool provides a list of all available configuration items.

extractListOfTools: The SEE extracts the list of registered SEE tools. For each tool the SEE requests the save all data operation.

Relates to Engineering Activity: 02. The SEE extracts the list of registered tools and requests each tool to save all its data.

processTools: Process the list of tools.

Relates to Engineering Activity: 03. The UAM requests from each tool a list of available configuration items.

removeUser: The SEE Administrator removes a user from the database.

Relates to Engineering Activity: 09. The SEE Administrator removes a user from the database.

removeUserGroup: The SEE Administrator user group from the data base.

Relates to Engineering Activity: 10. The SEE Administrator user group from the data base.

selectAccessForConfigItem: The user selects the access rights of the user group for the dedicated configuration item.

Relates to Engineering Activity: 05. The SEE Admin selects for each config item the access rights for the group.

storeSelectionInDB: UAM stores the access information for the dedicated configuration item in the data base.

Relates to Engineering Activity: 06. The UAM stores the access rights information in the database.

3.4.3.3 EM205_03_02 - Load Artefact

ID: EM205_03_02

Each tool of the engineering environment is requesting access information for the current logged in user from a central instance. The user related rights contain information

- if an artefact may be loaded and modified
- if an artefact may be loaded and modifications are not allowed
- if an artefact may not be loaded

The returned information depends on the access rights assigned to the group of users the current user is assigned to and on the change control information i.e. if change control is enforced and an approved change request is available for the requested artefact, where the current user is tasked to implement the change.

EM Purpose: The user loads an artefact of the activated baseline.

EM Pre-Condition: A dedicated baseline is activated from the current trunk or the work.

EM Post-Condition: Access to the artefact is denied if access is denied. The artefact is loaded in edit mode or in read mode.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The user requests to show all the available configuration items (or artefacts).
02. The user selects the configuration item he wants to load.
03. The SEE determines the active baseline for the given configuration item.
04. The tool checks the baseline and requests the access mode.
05. The SEE requests the access mode from the UAM tool for the currently logged in user.
06a. The UAM determines the access mode for the requested baseline of the configuration item.
06b. The UAM determines if the item is locked i.e. in use by other users.
07. The SEE checks the access mode and returns in case of access denied or access read only. If access is provided the SEE checks if changes are allowed.
08. The CM checks if for the given configuration item change control is activated.
09. The SEE returns modify access if change control is not yet activated otherwise it requests a list of open change requests.
10. The CM tool provides a list of all open CR for the requested config item.
11. The SEE checks open CR. If there are CR open the SEE returns modify access for the given configuration item. Otherwise, it returns read only access.
12. If the receives no access the tool does not load the baseline of the configuration item and informs the user.
13. If the tool receives read only access the tool loads the given baseline of the configuration item in read only mode.
14. The tool receives full access and loads the configuration item in read write mode.

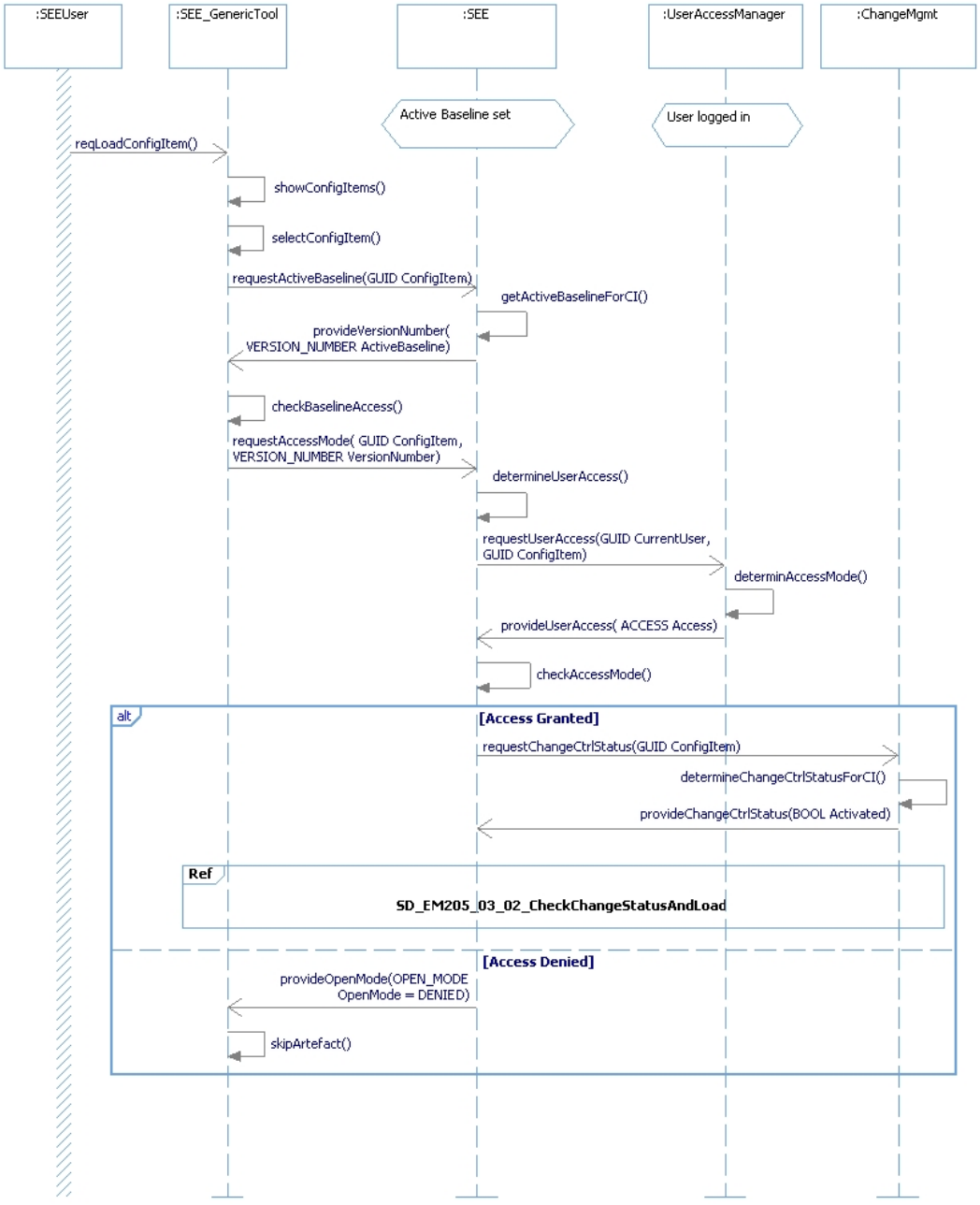

Figure 3-52 SD_EM205_03_02_LoadArtefact

checkAccessMode: If access to the SEE is denied the UAM provides the access mode ACCESS_DENIED to the tool.

Relates to Engineering Activity: 03. The SEE checks the access mode.

checkBaselineAccess: The tool need to check the access mode for the given version of the configuration item.

Relates to Engineering Activity: 04. The tool checks the baseline and requests the access mode.

determinAccessMode: The UAM determines for the given configuration item the general access rights for the logged in user.

Relates to Engineering Activity: 06a. The UAM determines the access mode for the requested baseline of the configuration item.

determineChangeCtrlStatusForCI: Change management checks if change control for the given configuration item is activated.

Relates to Engineering Activity: 08. The CM checks if for the given configuration item change control is activated.

determineUserAccess: The SEE as a central instance for automation determines the access rights for the given configuration item.

Relates to Engineering Activity: 05. The SEE requests the access mode from the UAM tool for the currently logged in user.

getActiveBaselineForCI: self-explanatory.

selectConfigItem: The user selects from the list of available configuration items the one, which he wants to open.

Relates to Engineering Activity: 02. The user selects the configuration item he wants to load.

showConfigItems: Show a list of config items to select from.

Relates to Engineering Activity: 01. The user requests to show all the available configuration items (or artefacts).

skipArtefact: Don't load the artefact and inform the user.

Relates to Engineering Activity: 12. If the receives no access the tool does not load the baseline of the configuration item and informs the user.

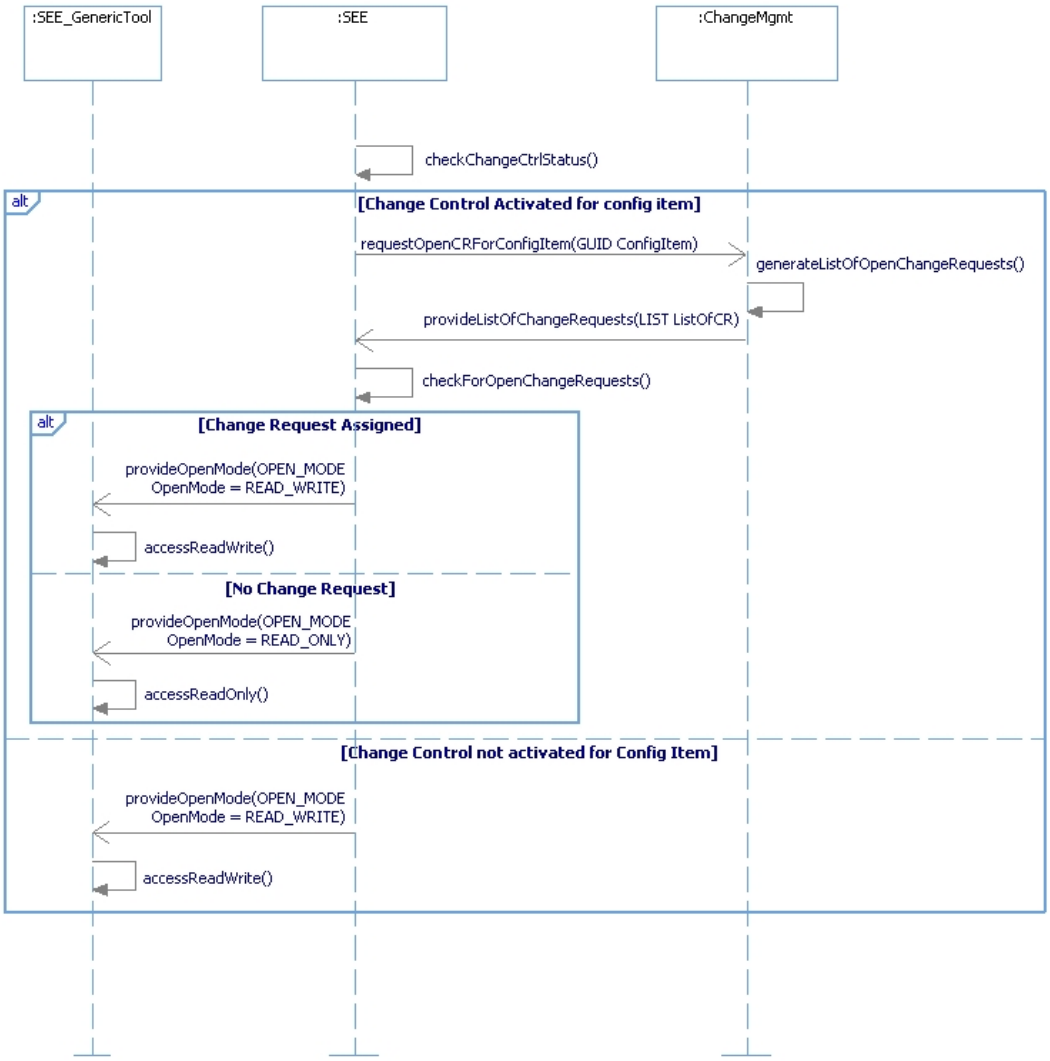

Figure 3-53 SD_EM205_03_02_CheckChangeStatusAndLoad

accessReadOnly: The tool loads the requested baseline for the selected configuration item in read only mode.

Relates to Engineering Activity: 13. If the tool receives read only access the tool loads the given baseline of the configuration item in read only mode.

accessReadWrite: The tool loads the requested baseline for the selected configuration item in read / write mode.

Relates to Engineering Activity: 14. The tool receives full access and loads the configuration item in read write mode.

checkChangeCtrlStatus: The SEE check the status of change control for the given configuration item.

Relates to Engineering Activity: 09. The SEE returns modify access if change control is not yet activated otherwise it requests a list of open change requests.

checkForOpenChangeRequests: The SEE checks open CR. If there are change requests assigned the SEE returns access to create an artefact for the parent configuration item. Otherwise, it returns no create access.

Relates to Engineering Activity: 11. The SEE checks open CR. If there are CR open the SEE returns access to create an artefact for the parent configuration item. Otherwise, it returns no create access.

generateListOfOpenChangeRequests: The change management extracts from the data base all open change requests assigned for the given configuration item. The change management provides a list of all change requests found.

Relates to Engineering Activity: 10. The CM tool provides a list of all open CR for the requested config item.

3.4.3.4 EM205_03_03 - Sign On SEE and create New Artefact

ID: EM205_03_03

Sign on to the SEE Environment and create a new artefact.

EM Purpose: The engineering method covers the sign on process to the SEE and the creation of a new artefact in a SEE tool.

EM Pre-Condition: The user is known in the UAM and allocated to a user group. The user is not logged in.

EM Post-Condition: The user is logged in to the SEE and a new artefact is created. The access rights for the artefact are assigned in the UAM.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

00. The requests to load a project. The SEE requests general access at the UAM.
01. The user signs in with user name and password at the SEE.
02. The UAM checks if the user is registered in its DB and provides the access status.
03. The SEE checks the access mode.
04. In case the user has no access, the SEE informs the user and quits the process.
05a. If access is granted the SEE sets the user as current user on the machine.
05b. If access is granted the SEE sets the user as current user on the machine.
06. The SEE informs the user about successful sign on.
07. The SEE checks the access mode and returns in case of access denied. If access is provided the SEE checks if changes are allowed.
07. The SEEUser requests to create a new artefact in a tool. The tool requests create access.
08. The CM checks if for the given configuration item change control is activated.
08. The SEE processes the create access request and requests create access mode for the current user from UAM.
09. The SEE returns modify access if change control is not yet activated otherwise it requests a list of open change requests.
09. The UAM checks if the given user has a right to create an artefact in the given tool.
10. The CM tool provides a list of all open CR for the requested config item.
11. The SEE checks open CR. If there are CR open the SEE returns access to create an artefact for the parent configuration item. Otherwise, it returns no create access.
12. The tool quits the operation if create access is not provided.
13. The tool creates the artefact and registers the config item at the UAM.

14. The UAM adds the new configuration item to its database.
15. The UAM assigns the access rights to new configuration item (different policies possible) and reports the status.
16. The tool opens the artefact in edit mode.

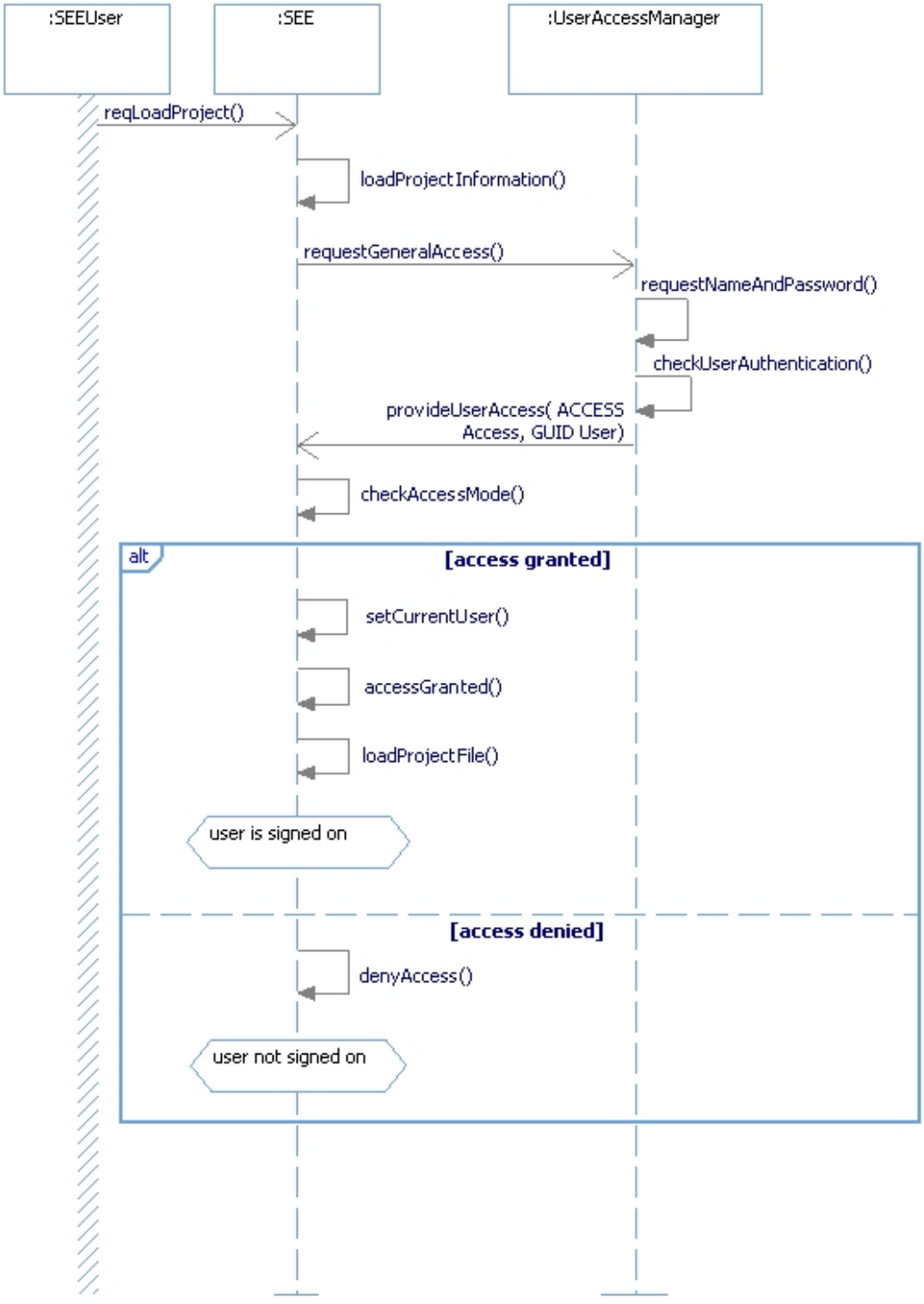

Figure 3-54 SD_EM205_03_03_SignOn

accessGranted: The user is granted access to the database of all the registered tools.

Relates to Engineering Activity: 06. The SEE informs the user about successful sign on.

checkAccessMode: If access to the SEE is denied the UAM provides the access mode ACCESS_DENIED to the tool.

Relates to Engineering Activity: 03. The SEE checks the access mode.

checkUserAuthentication: The UAM checks if the user is registered in the database. If the user is found and the password is correct it provides the status ACCESS_GRANTED, else it provides the status ACCESS_DENIED.

Relates to Engineering Activity: 02. The UAM checks if the user is registered in its DB and provides the access status.

denyAccess: The user is not registered in the database. Inform the user and quit.

Relates to Engineering Activity: 04. In case the user has no access, the SEE informs the user and quits the process.

loadProjectFile: Load all data from the project file.

Relates to Engineering Activity: 05b. If access is granted the SEE sets the user as current user on the machine.

loadProjectInformation: The user requests to load a project. The SEE loads information from the project file and requests general access to the project for the current user and the local machine.

Relates to Engineering Activity: 00. The requests to load a project. The SEE requests general access at the UAM.

requestNameAndPassword: The SEEUser logs in at the SEE at the current machine. Name and Password are one possibility for authentication. Others could be provided as well.

Relates to Engineering Activity: 01. The user signs in with user name and password at the SEE.

setCurrentUser: The UAM set the given user as the current user on the machine.

Relates to Engineering Activity: 05a. If access is granted the SEE sets the user as current user on the machine.

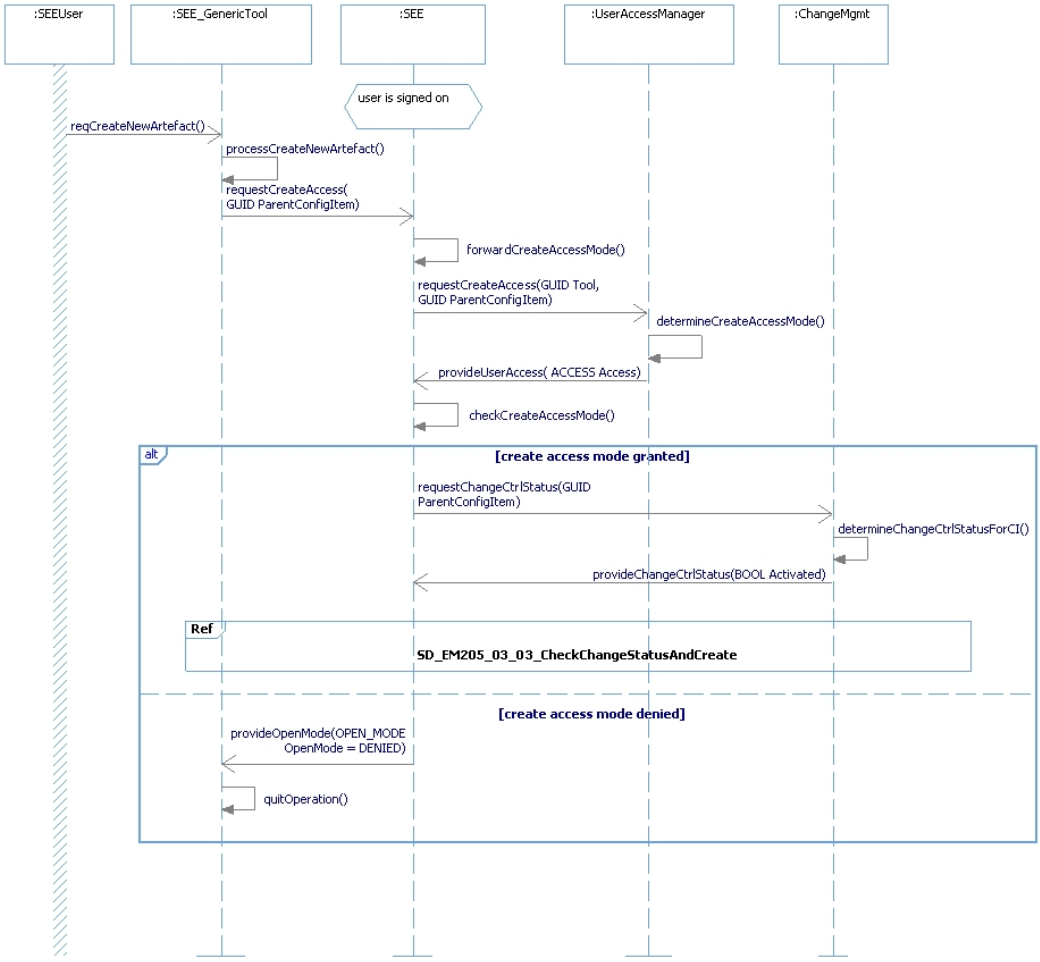

Figure 3-55 SD_EM205_03_03_CreateArtefact

checkCreateAccessMode: The SEE checks the received information.

Relates to Engineering Activity: 07. The SEE checks the access mode and returns in case of access denied. If access is provided the SEE checks if changes are allowed.

determineChangeCtrlStatusForCI: Change management checks if change control for the given configuration item is activated.

Relates to Engineering Activity: 08. The CM checks if for the given configuration item change control is activated.

determineCreateAccessMode: The UAM checks if the given user has the right to create new artefacts in the scope of the provided parent.

Relates to Engineering Activity: 09. The UAM checks if the given user has a right to create an artefact in the given tool.

forwardCreateAccessMode: The SEE processes forwards create access request and requests create access mode for the current user from UAM.

Relates to Engineering Activity: 08. The SEE processes the create access request and requests create access mode for the current user from UAM.

processCreateNewArtefact: self-explanatory.

quitOperation: Quit the operation.

Relates to Engineering Activity: 12. The tool quits the operation if create access is not provided.

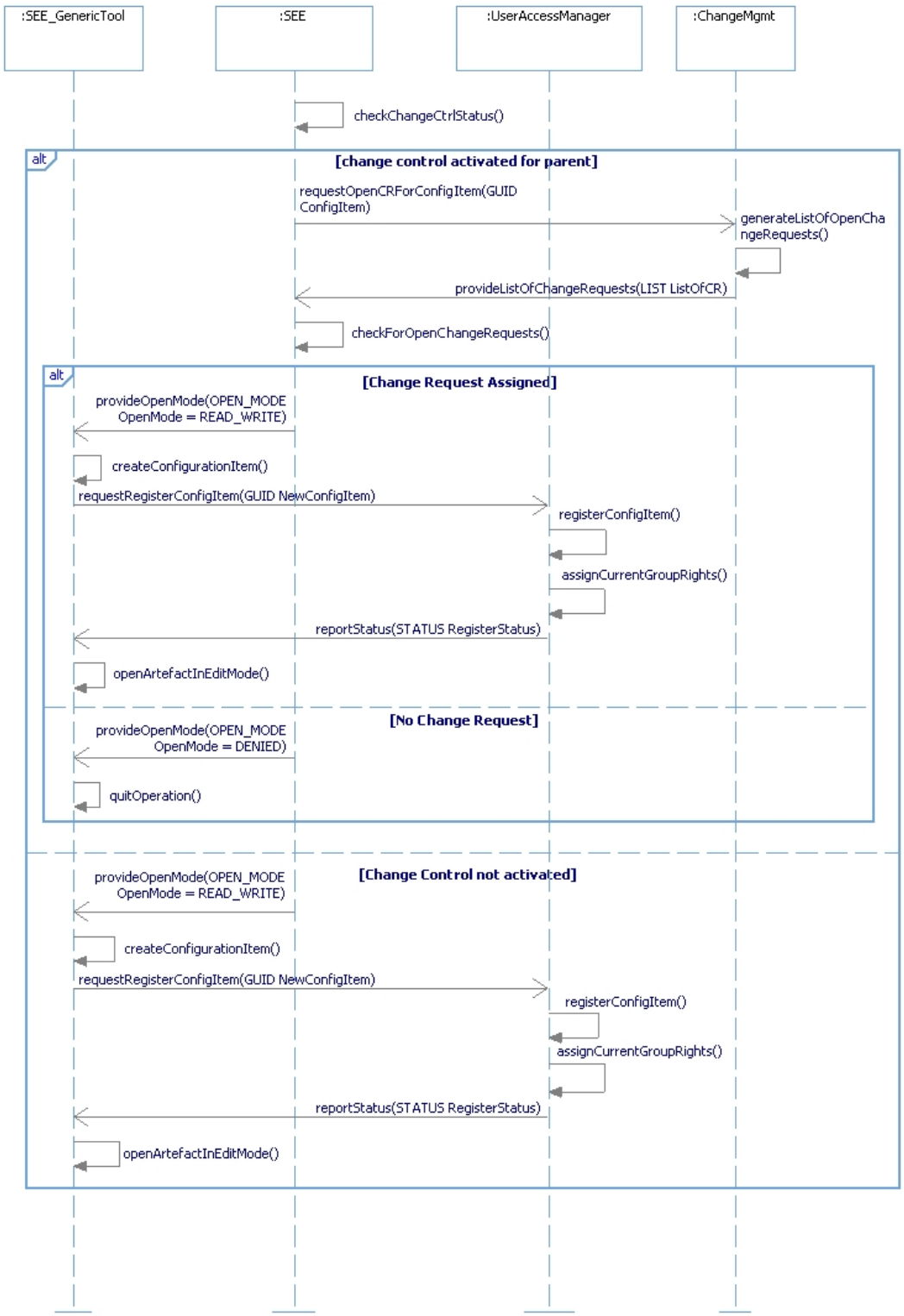

Figure 3-56 SD_EM205_03_03_CheckChangeStatusAndCreate

assignCurrentGroupRights: The UAM assigns the access rights to the new configuration item and reports the status. Different policies are possible e.g. to derive the permissions from the parent item.

Relates to Engineering Activity: 15. The UAM assigns the access rights to new configuration item (different policies possible) and reports the status.

checkChangeCtrlStatus: The SEE check the status of change control for the given configuration item.

Relates to Engineering Activity: 09. The SEE returns modify access if change control is not yet activated otherwise it requests a list of open change requests.

checkForOpenChangeRequests: The SEE checks open CR. If there are change requests assigned the SEE returns access to create an artefact for the parent configuration item. Otherwise, it returns no create access.

Relates to Engineering Activity: 11. The SEE checks open CR. If there are CR open the SEE returns access to create an artefact for the parent configuration item. Otherwise, it returns no create access.

createConfigurationItem: Create the intended configuration item and request to register it at the UAM.

Relates to Engineering Activity: 13. The tool creates the artefact and registers the config item at the UAM.

generateListOfOpenChangeRequests: The change management extracts from the data base all open change requests assigned for the given configuration item. The change management provides a list of all change requests found.

Relates to Engineering Activity: 10. The CM tool provides a list of all open CR for the requested config item.

openArtefactInEditMode: Load the newly created artefact in edit mode.

Relates to Engineering Activity: 16. The tool opens the artefact in edit mode.

quitOperation: Quit the operation.

Relates to Engineering Activity: 12. The tool quits the operation if create access is not provided.

registerConfigItem: The UAM registers the configuration item in its data base.

Relates to Engineering Activity: 14. The UAM adds the new configuration item to its database.

3.4.4 PA205_04 - Process Automation

ID: PA205_04

Related User Story: Process Automation, Guidance and Monitoring

Automation of activities often performed by the engineer or common to the tools.

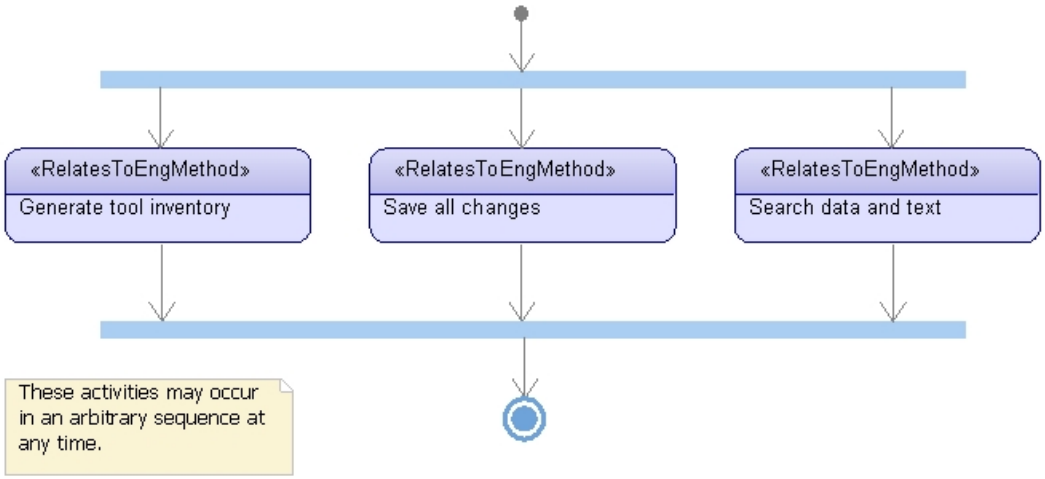

Figure 3-57 AD_PA205_04

Generate tool inventory (RelatesToEngMethod): Generate the tool inventory to complete the documentation package for a certain product release. The inventory of the tools list the tools and their revision installed in the SEE and used for the generation of the data package. This is important to be later on able to re-install the tool environment.

Save all changes (RelatesToEngMethod): One of certainly several functions supporting the acceptance of the SEE. The "save all" function is triggered at one tool and the SEE assures that all the other installed tools are also saving their data. This assures consistent SEE data in all the connected tools.

If working with a single tool the user is often not aware where data is changed in the background due to IOS tool interaction.

Search data and text (RelatesToEngMethod): Searching the data of the different tool need to be supported from a single place.

3.4.4.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

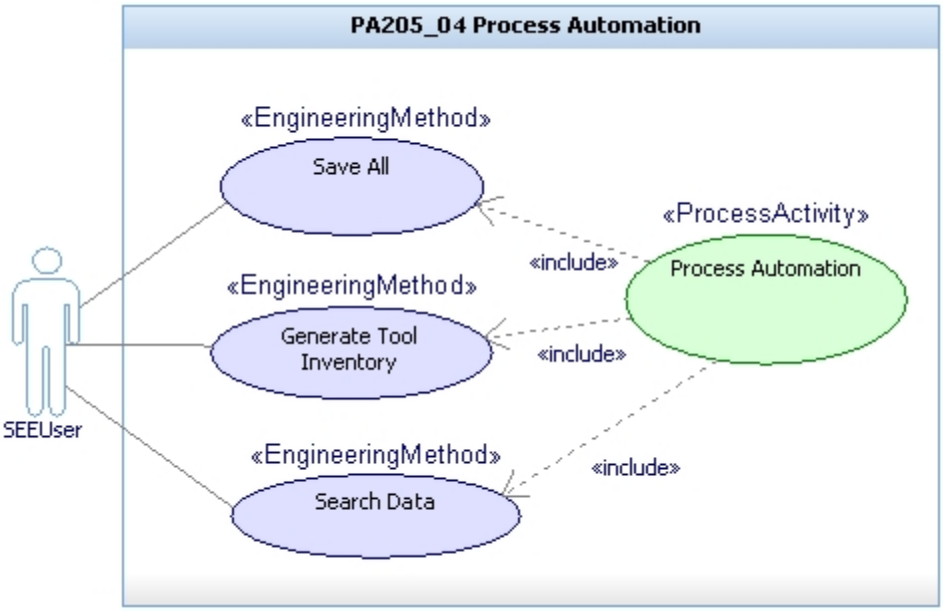

Figure 3-58 UCD_GeneralUseCases

3.4.4.2 EM205_04_01 - Generate Tool Inventory

ID: EM205_04_01

Generate a list of tools and their version information.

EM Purpose: Generates an inventory list of all the tools registered with the SEE providing detailed version information and configuration.

EM Pre-Condition: Tools are registered in the SEE

EM Post-Condition: N/A

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The user requests the SEE to generate an inventory list.
02. The SEE generates a report and requests each SEE tool to provide its version information
03. The SEE tool extracts the version information and provides the information to the SEE.
04. The SEE collects the version information from all the SEE tools.
05. The SEE writes the received version information to the report.

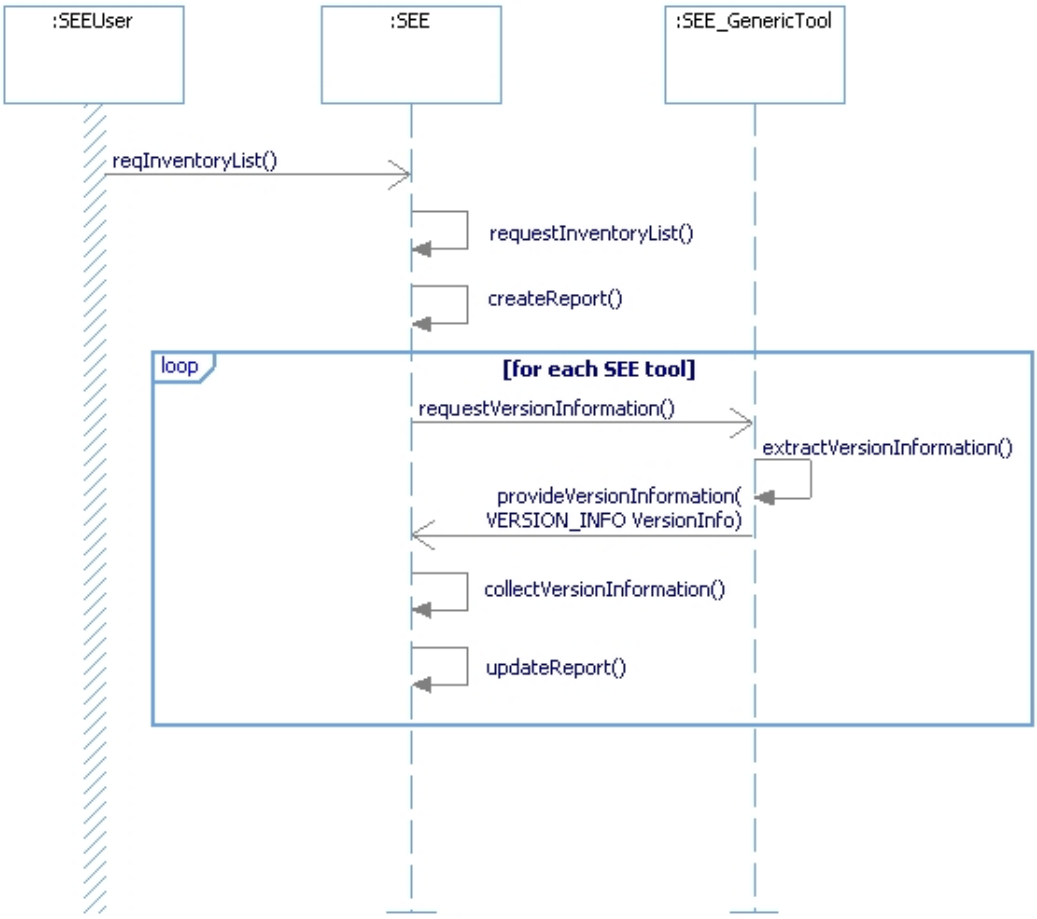

Figure 3-59 SD_EM205_04_01_GenerateToolInventory

collectVersionInformation: The SEE collects the version information from all the SEE tools.

Relates to Engineering Activity: 04. The SEE collects the version information from all the SEE tools.

createReport: The SEE generates a report and requests each SEE tool to provide the version information.

Relates to Engineering Activity: 02. The SEE generates a report and requests each SEE tool to provide its version information

extractVersionInformation: The SEE tool extracts the version information and provides the information to the SEE.

Relates to Engineering Activity: 03. The SEE tool extracts the version information and provides the information to the SEE.

requestInventoryList: self-explanatory.

updateReport: The SEE writes the received version information to the report.

Relates to Engineering Activity: 05. The SEE writes the received version information to the report.

3.4.4.3 EM205_04_02 - Save All

ID: EM205_04_02

Saves all the changes made to the current system design work stream nodes across all tools.

EM Purpose: Save all the changed data throughout the SEE for a consistent view.

EM Pre-Condition: A project is loaded into the workspace and changes are performed.

EM Post-Condition: All modifications across all the tools are saved to the database.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The user selects the save all operation at some tool. The tool forwards the request to the SEE.
02. The SEE extracts the list of registered tools and requests each tool to save all its data.
03. The tool save all its data and provides an operation status.
04. The SEE collects the status of each tool and generates a combined one.
05. The triggering tool displays the combined status.
28. The change management tool requests a list of selected configuration items from the SEE.

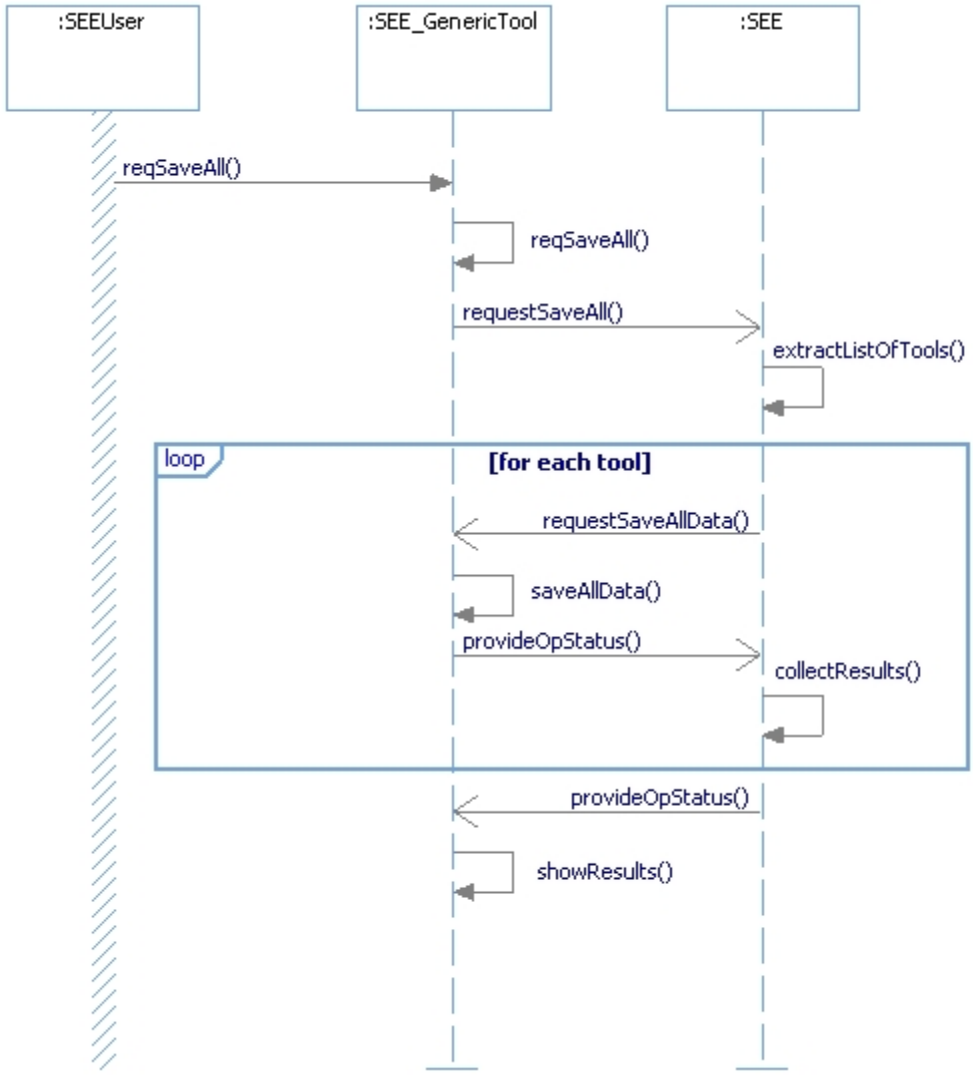

Figure 3-60 SD_EM205_04_02_SaveAll

collectResults: The SEE collects the status of the save all operation at each tool and provides a combined status.

Relates to Engineering Activity: 04. The SEE collects the status of each tools and generates a combined one.

extractListOfTools: The SEE extracts the list of registered SEE tools. For each tool the SEE requests the save all data operation.

Relates to Engineering Activity: 02. The SEE extracts the list of registered tools and requests each tool to save all its data.

reqSaveAll: The user selects the save all operation at some tool in the SEE. The tool requests the save all operation at the SEE.

Relates to Engineering Activity: 01. The user selects the save all operation at some tool. The tool forwards the request to the SEE.

saveAllData: The tool saves all data and provides the status of the operation.

Relates to Engineering Activity: 03. The tool save all its data and provides an operation status.

showResults: The tool displays the collected status.

Relates to Engineering Activity: 05. The triggering tool displays the combined status.

3.4.4.4 EM205_04_03 - Search Data

ID: EM205_04_03

Search text, artefacts and model elements in the SEE Environment.

EM Purpose: To search the data across the SEE tools.

EM Pre-Condition: N/A

EM Post-Condition: N/A

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. The user requests to search for elements including a given search string.
02. The user defines the search string using regular expressions.
03. The SEE tool searches the database for occurrences of the given search string and provides a list of all findings.
04. The SEE collect the findings.
05. The SEE displays all the findings.
06. The user is interested in which tool, configuration item and baseline a dedicated element is included.
07. The user defines the ID optionally using wildcards.
08. The SEE tool searches the database and all baselines for occurrences of an element with the given ID.
09. The SEE collect the findings.
10. The SEE displays all the findings.

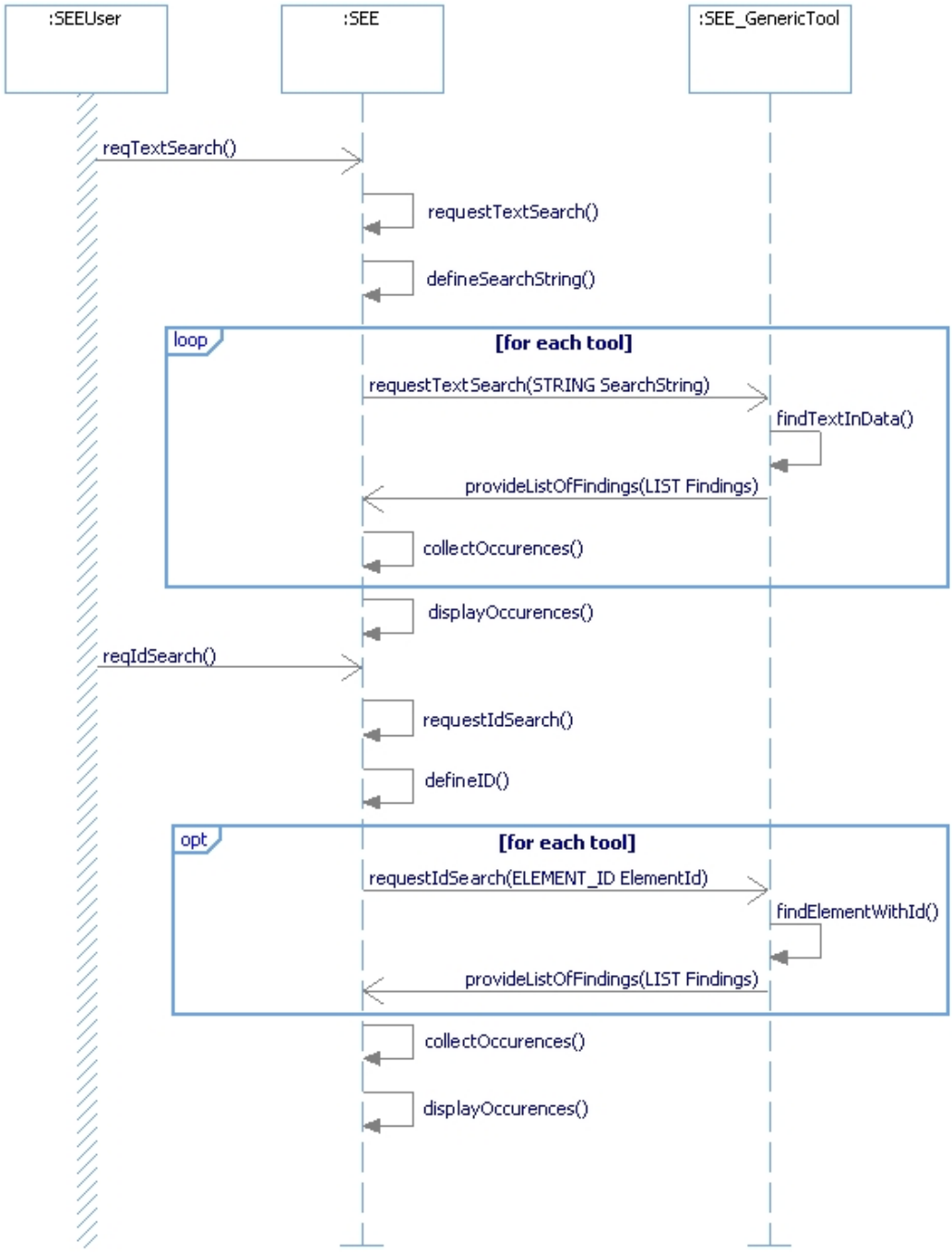

Figure 3-61 SD_EM205_04_03_SearchData

collectOccurences: The SEE collect the findings.

Relates to Engineering Activity: 09. The SEE collect the findings.

defineID: The user defines the ID optionally using wildcards. The ID includes the GUID of the configuration item, the GUID of the baseline and the index of the element within the configuration item. Parts of the ID can be replaced by wildcards. In this case it is possible to search all the baselines of a dedicated configuration item.

Relates to Engineering Activity: 07. The user defines the ID optionally using wildcards.

defineSearchString: The user defines the search string using regular expressions.

Relates to Engineering Activity: 02. The user defines the search string using regular expressions.

displayOccurences: The SEE displays all the findings.

Relates to Engineering Activity: 10. The SEE displays all the findings.

findElementWithId: The SEE tool searches the database and all baselines for occurrences of an element with the given ID. The ID includes the GUID of the configuration item, the GUID of the baseline and the index of the element within the configuration item.

Relates to Engineering Activity: 08. The SEE tool searches the database and all baselines for occurrences of an element with the given ID.

findTextInData: The SEE tool searches the database for occurrences of the given search string and provides a list of all findings. The findings include a reference to the element.

Relates to Engineering Activity: 03. The SEE tool searches the database for occurrences of the given search string and provides a list of all findings.

requestIdSearch: The user is interested in which tool, configuration item and baseline a dedicated element is included.

Relates to Engineering Activity: 06. The user is interested in which tool, configuration item and baseline a dedicated element is included.

requestTextSearch: The user requests to search for elements including a given search string.

Relates to Engineering Activity: 01. The user requests to search for elements including a given search string.

3.5 User Story US206 - Project compliance monitoring based on advanced traceability

The purpose of the user story ”Project Compliance Monitoring based on Advanced Traceability” is to set up a full traceability between the artefacts of the product development. This includes:

- the specification of a traceability meta model
- the creation of requirements traceability links
- the analysis of artefacts using traceability links (validation, verification, change management)
- and the support of certification by traceability-based project monitoring.

The Creation of the Traceability for User Story "Project compliance monitoring based on advanced traceability" is demonstrated in the process activity "Perform Verification". The creation of traceability for requirements, design, safety and so on is covered in the other user stories. Traceability between artefacts not in the scope of the use case WP203 are not considered.

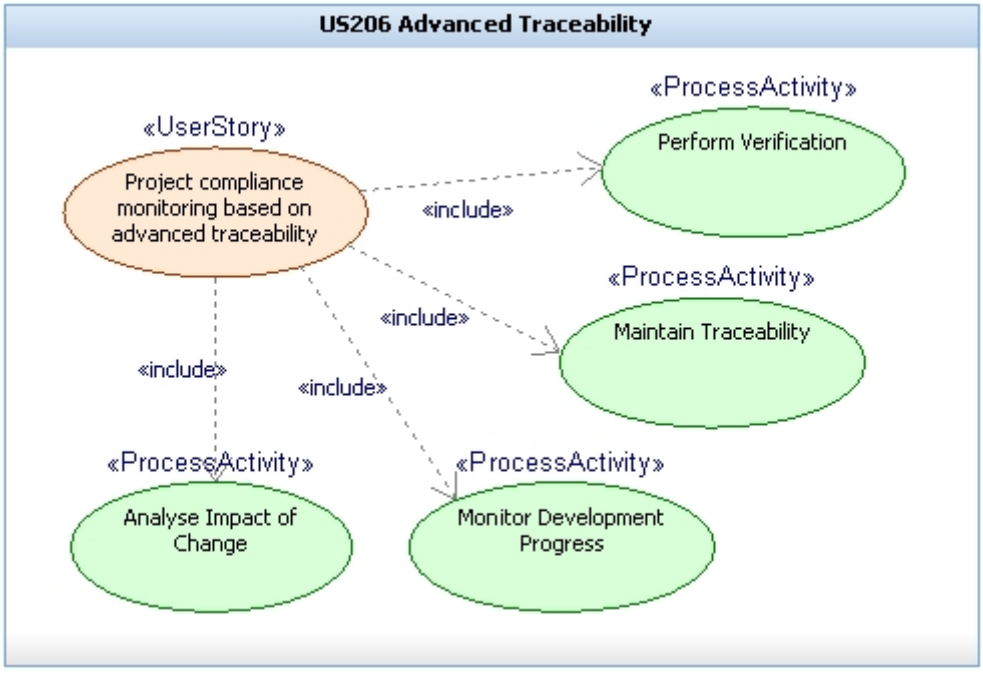

Figure 3-62 UCD_US206_AdvancedTraceability

The User Story Project compliance monitoring based on advanced traceability is including the Process Activities listed below. Note, that a Process Activity is detailed within another User Story if no <<include>> dependency is established from the User Story.

- Maintain Traceability
- Monitor Development Progress
- Perform Verification

- Analyse Impact of Change

3.5.1 PA206_01 - Maintain Traceability

ID: PA206_01

Related User Story: Project compliance monitoring based on advanced traceability

The purpose of maintain traceability is to identify artefacts and links that need to be modified because the source of the linked artefact was changed.

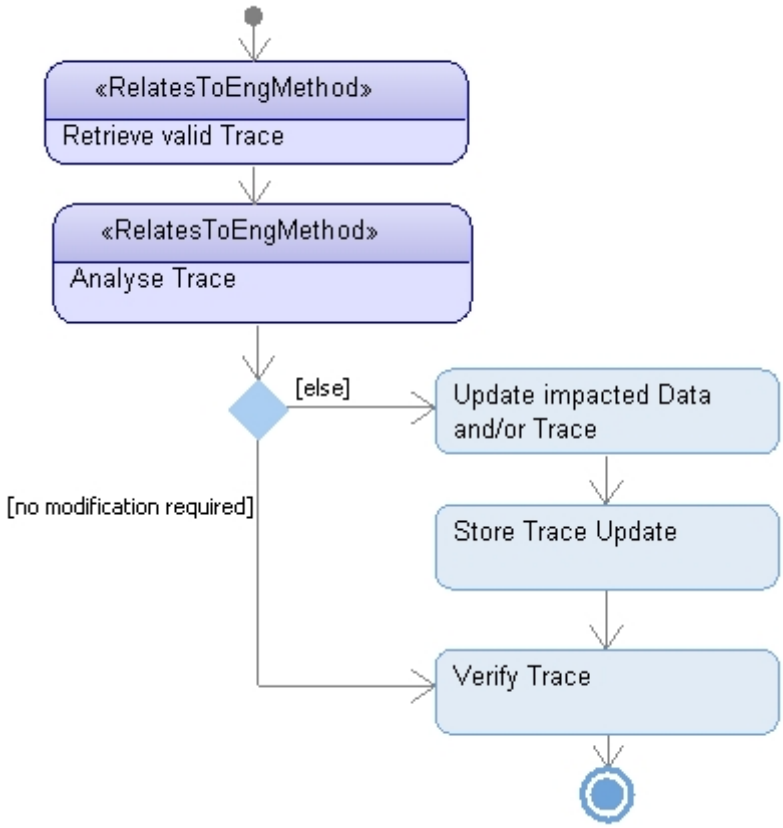

Figure 3-63 AD_PA206_01

Analyse Trace (RelatesToEngMethod): Analyse the trace to be maintain according to the changes of its target artefact and change the status of the trace to suspect

Retrieve valid Trace (RelatesToEngMethod): Retrieve the valid trace to be maintained due to changes in the source and/or destination artefact

Store Trace Update: Store executed update

Update impacted Data and/or Trace: Update the impacted data in the source or destination artefact or delete or update the trace

Verify Trace: Verify that the trace is consistent and if appropriate updated correctly.

3.5.1.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

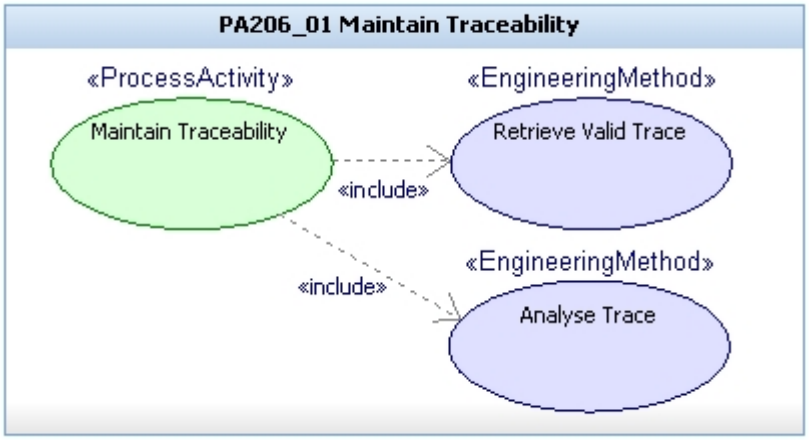

Figure 3-64 UCD_MaintainTraceability

3.5.1.2 EM206_01_01 - Retrieve Valid Trace

ID: EM206_01_01

Retrieve the valid trace to be maintained. A trace is valid if the source artefact has the status valid.

EM Purpose: To get the valid links of the system engineering environment for accessibility.
EM Pre-Condition: Links of valid artefacts are available in the system engineering environment.
EM Post-Condition: Valid Links are accessible in the system engineering environment.
List of related Engineering Activities:
The Engineering Method is described by the following Engineering Activities:

01. Requests for the outgoing links of all artefacts, of which the status is valid
02. Illustrate the traceability network within SEE by displaying the artefacts and the corresponding links

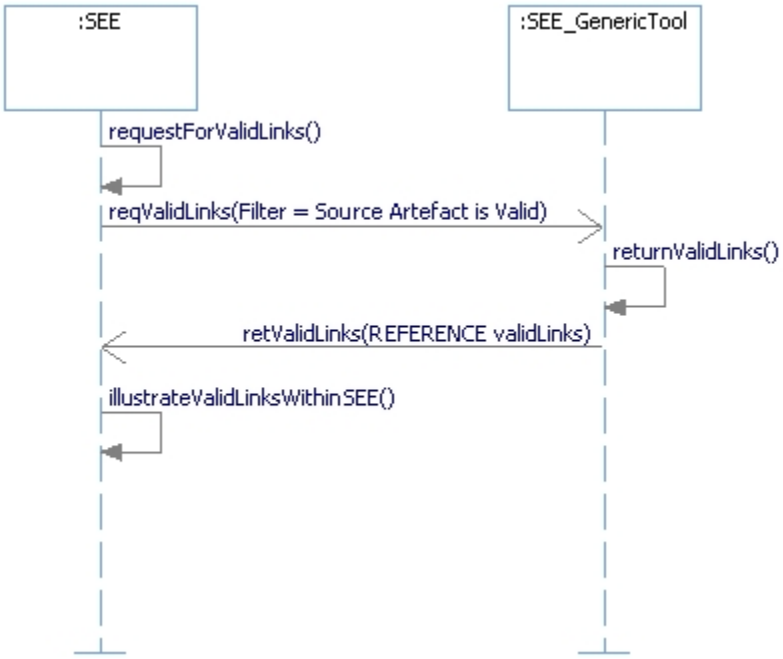

Figure 3-65 SD_EM206_01_01_Retrieve valid Trace

illustrateValidLinksWithinSEE: Illustrate the traceability network within SEE by displaying the artefacts and the corresponding links.

Relates to Engineering Activity: 02. Illustrate the traceability network within SEE by displaying the artefacts and the corresponding links

requestForValidLinks: To get the valid links from all available links in the environment, the SEE requests for the outgoing links of all artefacts, of which the status is valid.

Relates to Engineering Activity: 01. Requests for the outgoing links of all artefacts, of which the status is valid

returnValidLinks: Return the valid outgoing links of including artefacts.

3.5.1.3 EM206_01_02 - Analyse Trace

ID: EM206_01_02

Analyse the trace to be maintain according to the changes of its target artefact and change the status of the trace to suspect

EM Purpose: To analyse the environment for suspect links and mark them respectively.

EM Pre-Condition: Valid Links are accessible in the system engineering environment.

EM Post-Condition: Links with changed target artefact were marked as suspect.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Analyse if artefacts within SEE were changed
02. Check for links, which are affected by a changed target
03. Mark the affected links as suspect
04. Within the SEE the suspected links are highlighted to be identifiable by the user

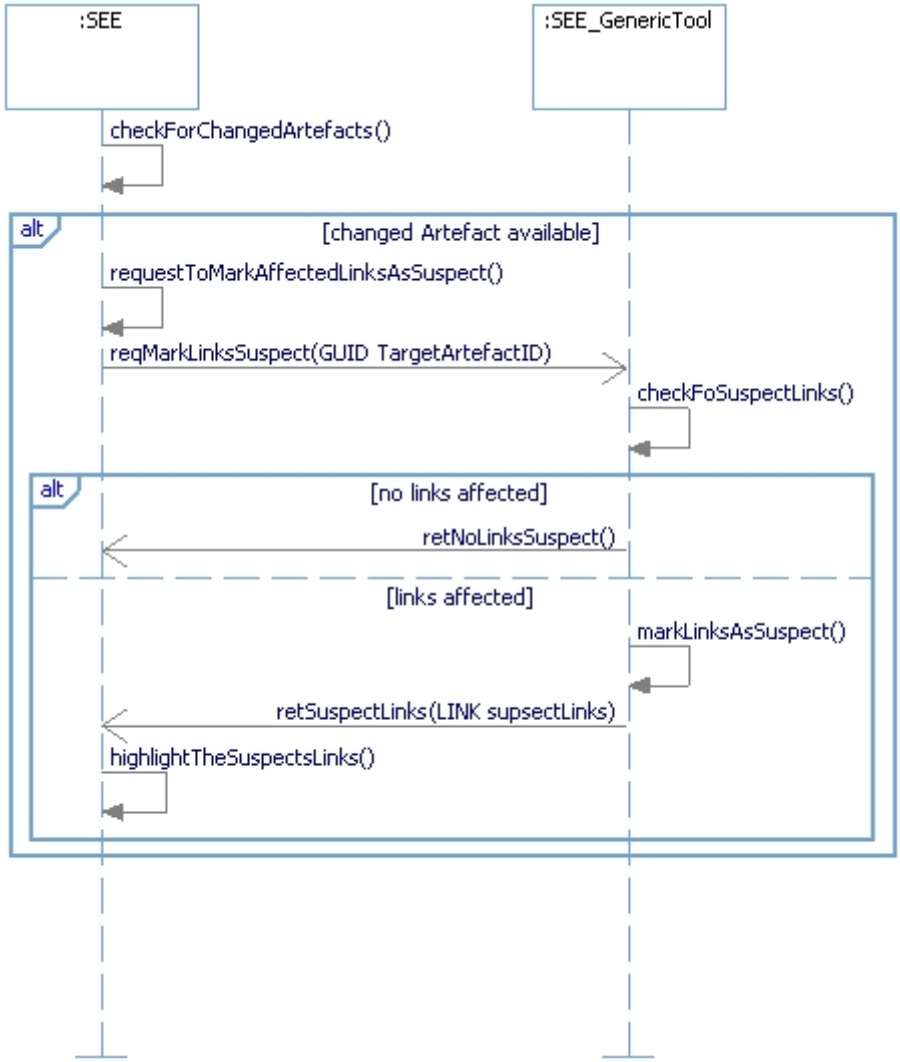

Figure 3-66 SD_EM206_01_02_Analyse Trace

checkForChangedArtefacts: Analyse if artefacts within SEE were changed

Relates to Engineering Activity: 01. Analyse if artefacts within SEE were changed

checkFoSuspectLinks: Check for links, which are affected by a changed target.

Relates to Engineering Activity: 02. Check for links, which are affected by a changed target

highlightTheSuspectsLinks: Within the SEE the suspected links are highlighted to be identifiable by the user.

Relates to Engineering Activity: 04. Within the SEE the suspected links are highlighted to be identifiable by the user

markLinksAsSuspect: Mark the affected links as suspect.

Relates to Engineering Activity: 03. Mark the affected links as suspect

requestToMarkAffectedLinksAsSuspect: request that all links, which have a changed artefact as target are marked as suspect.

3.5.2 PA206_02 - Monitor Development Progress

ID: PA206_02

Related User Story: Project compliance monitoring based on advanced traceability

Perform analysis for development process such as coverage analyses to ensure that all requirements are met in the functional analysis and covered by verification cases as well as compliance analysis to document that the requirements have been satisfied by the functional system model.

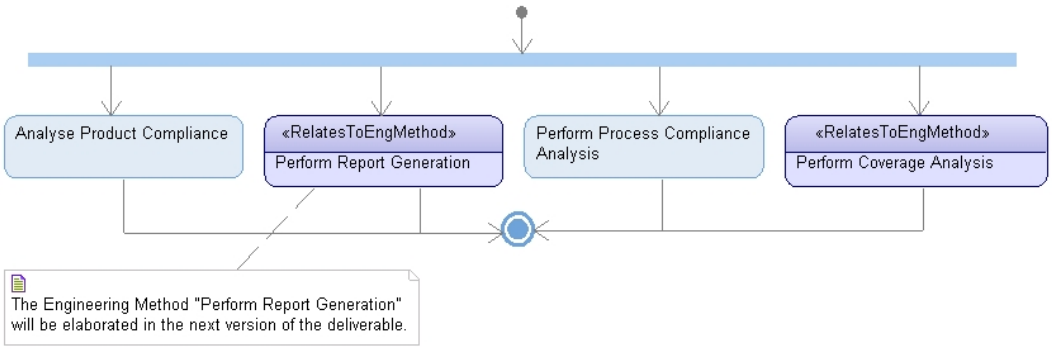

Figure 3-67 AD_PA206_02

Analyse Product Compliance: Perform evidence based on the traceability that the product meets the requirement of the stakeholders.

Perform Coverage Analysis (RelatesToEngMethod): Perform coverage analysis based on the link types as input for certification evidences, Key Performance Indicators (KPI) and so on.

Perform Process Compliance Analysis: Provide confirmation that the development process steps such as reviews and tests have been performed and the required documents have been provided.

Perform Report Generation (RelatesToEngMethod): Perform report generation to provide documents, e.g. for certification evidences, based on the established traceability.

3.5.2.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

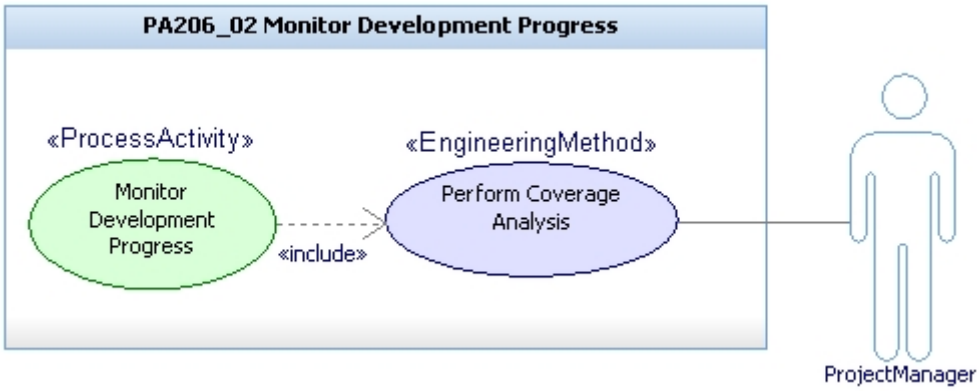

Figure 3-68 UCD_MonitorDevelopmentProgress

3.5.2.2 EM206_02_01 - Perform Coverage Analysis

ID: EM206_02_01

Perform coverage analysis based on the link types as input for certification evidences, Key Performance Indicators (KPI) and so on.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. get the list of configuration item for the selected baseline set.
02. Analyse the provided requirements for in-going links of the type refine and decompose.
03. Analyse the requirements of the provided requirements for in-going links of the type satisfy.
04. Analyse the requested requirements for the out-going links of the type refine and decompose.
05. Analyse the provided requirements for in-going links of the type verify.
06. Analyse the design artefacts for out-going links of the type satisfy.
07. Analyse the design artefacts for in-going links of the type derive.
08. Analyse the change actions for in-going links of the type modify.
09. Collect and prepare the results of the performed coverage analyses.
10. provide the coverage results.

To increase the readability this Engineering Method is partitioned into several sequence diagrams. The first diagram defines the main diagram. The subsequent diagrams describe parts of the main diagram, which are assigned within the sequence diagram with the "Ref"-Operator.

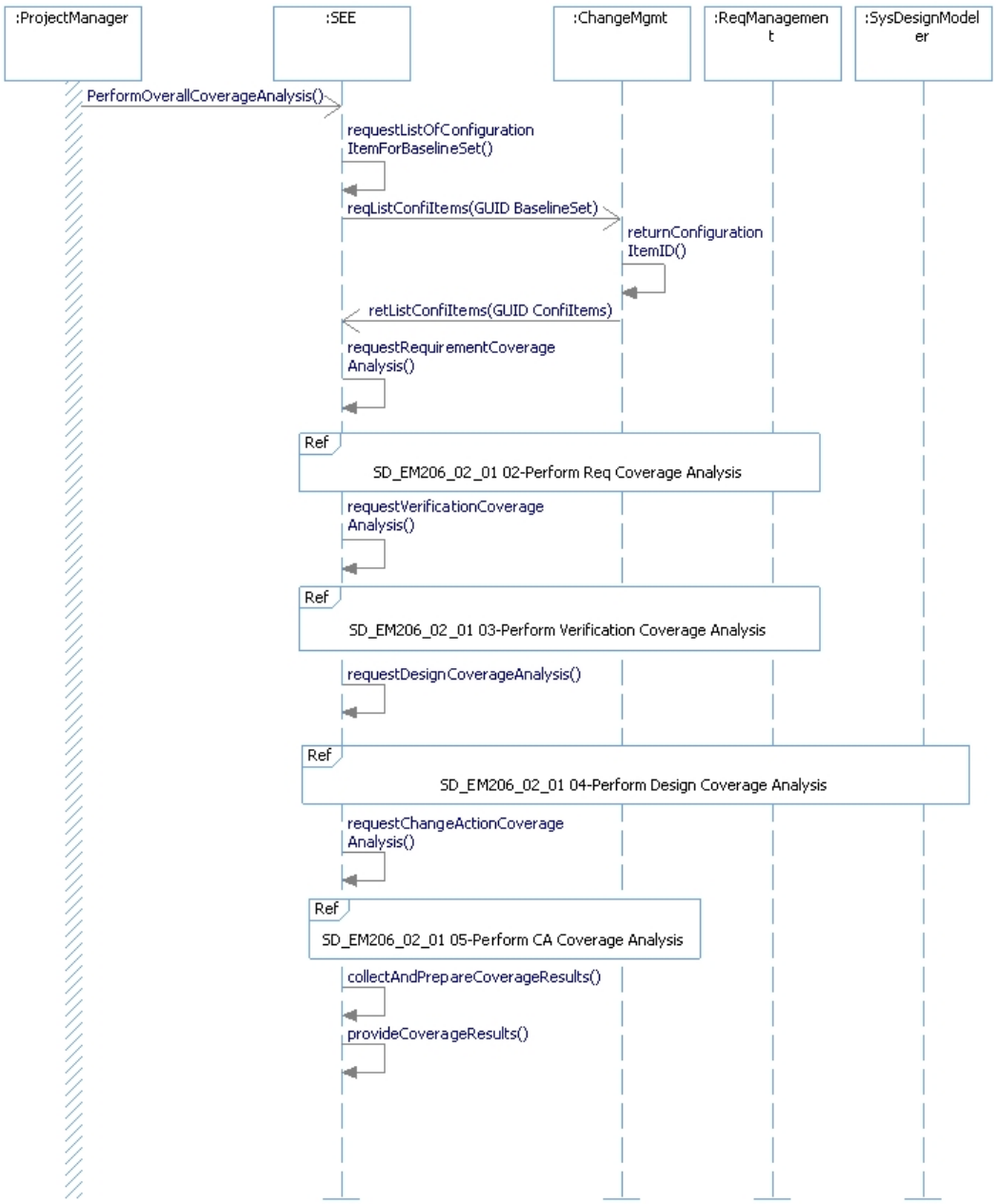

Figure 3-69 SD_EM206_02_01 01-Perform Coverage Analysis

collectAndPrepareCoverageResults: Collect and prepare the results of the performed coverage analyses.

Relates to Engineering Activity: 09. Collect and prepare the results of the performed coverage analyses.

provideCoverageResults: Provide the coverage results, which consist of a coverage overview in per cent and a coverage table containing as a minimum the artefact ID, the allocated artefact IDs and its results.

Relates to Engineering Activity: 10. provide the coverage results.

requestChangeActionCoverageAnalysis: Request change action coverage analysis based on the links of the type modify.

requestDesignCoverageAnalysis: Request coverage analysis of the design artefacts based on the links of the type satisfy and derive.

requestListOfConfigurationItemForBaselineSet: Get the list of configuration item for the selected baseline set.

Relates to Engineering Activity: 01. get the list of configuration item for the selected baseline set.

requestRequirementCoverageAnalysis: Request a top-down and bottom-up coverage analysis for requirements of a selected list regarding the in-going links of the type refine, decompose and satisfy.

requestVerificationCoverageAnalysis: Request verification coverage analysis for a set of requirements based on the in-going links of the type verify.

returnConfigurationItemID: Return the IDs of the configuration items within the defined baseline set.

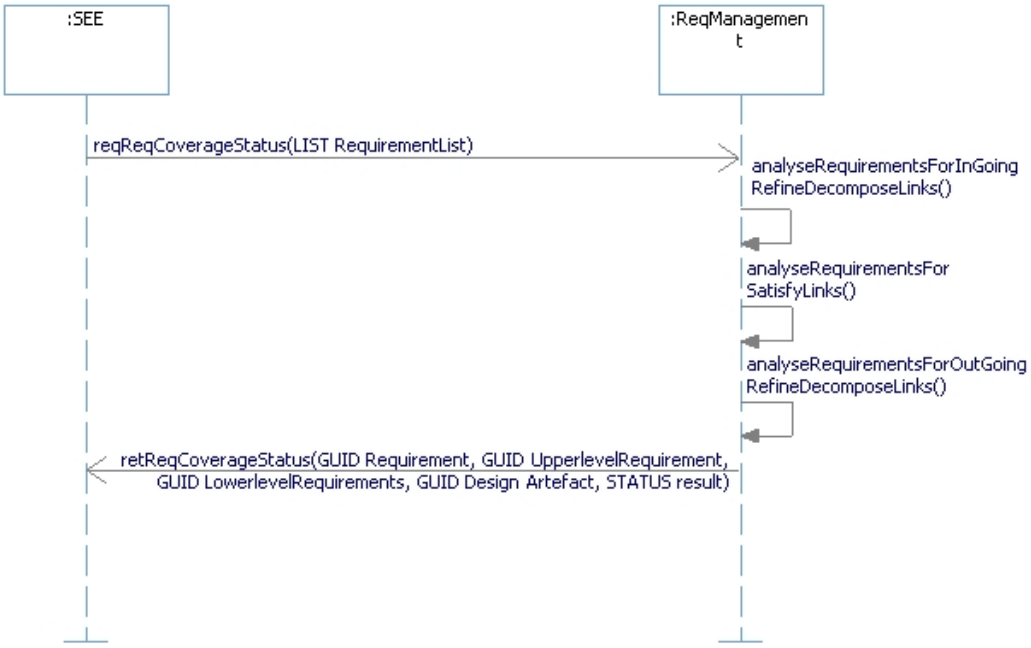

Figure 3-70 SD_EM206_02_01 02-Perform Req Coverage Analysis

analyseRequirementsForInGoingRefineDecomposeLinks: Analyse the requirements of the provided list for in-going links of the type refine and decompose.

Relates to Engineering Activity: 02. Analyse the provided requirements for in-going links of the type refine and decompose.

analyseRequirementsForOutGoingRefineDecomposeLinks: Analyse the bottom-up coverage for the requirements of the selected list regarding the out-going links of the type refine and decompose.

Relates to Engineering Activity: 04. Analyse the requested requirements for the out-going links of the type refine and decompose.

analyseRequirementsForSatisfyLinks: Analyse the requirements of the provided list for in-going links of the type satisfy.

Relates to Engineering Activity: 03. Analyse the requirements of the provided requirements for in-going links of the type satisfy.

Figure 3-71 SD_EM206_02_01 03-Perform Verification Coverage Analysis

analyseRequirementsForVerfyLinks: Analyse the requirements within the requested set of requirements for in-going links from the type verify.

Relates to Engineering Activity: 05. Analyse the provided requirements for in-going links of the type verify.

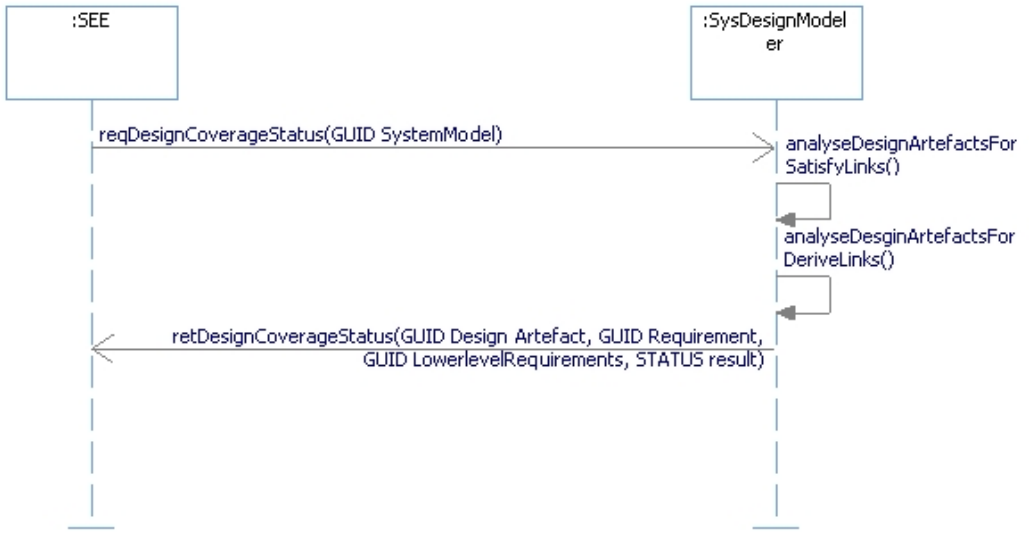

Figure 3-72 SD_EM206_02_01 04-Perform Design Coverage Analysis

analyseDesginArtefactsForDeriveLinks: Analyse the requirements of the provided list for in-going links of the type derive.

Relates to Engineering Activity: 07. Analyse the design artefacts for in-going links of the type derive.

analyseDesignArtefactsForSatisfyLinks: Analyse the design artefacts of the provided list for out-going links of the type satisfy.

Relates to Engineering Activity: 06. Analyse the design artefacts for out-going links of the type satisfy.

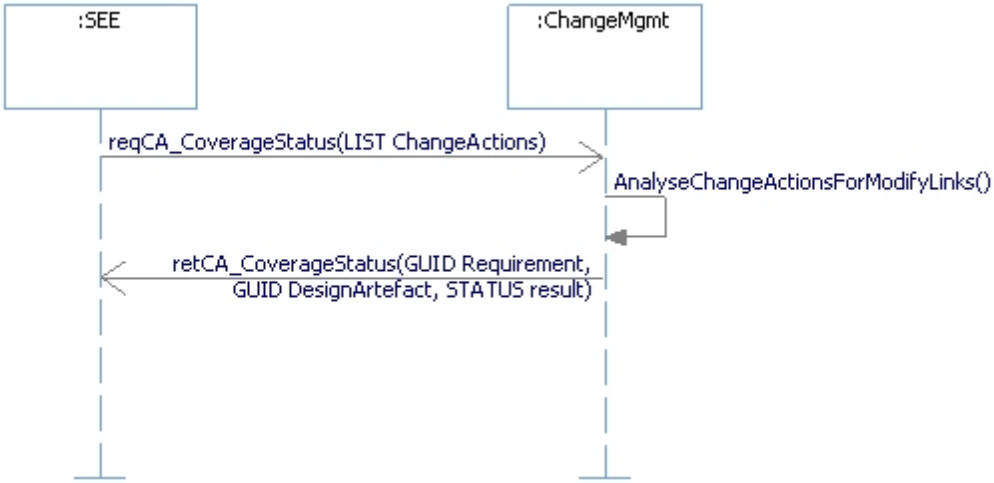

Figure 3-73 SD_EM206_02_01 05-Perform CA Coverage Analysis

AnalyseChangeActionsForModifyLinks: Analyse coverage of the change actions according to the links of the type modify.

Relates to Engineering Activity: 08. Analyse the change actions for in-going links of the type modify.

3.5.3 PA206_03 - Perform Verification

ID: PA206_03

Related User Story: Project compliance monitoring based on advanced traceability

The scope of this process activity is to trace artefacts from the requirements analysis and functional analysis to system verification objectives and cases.

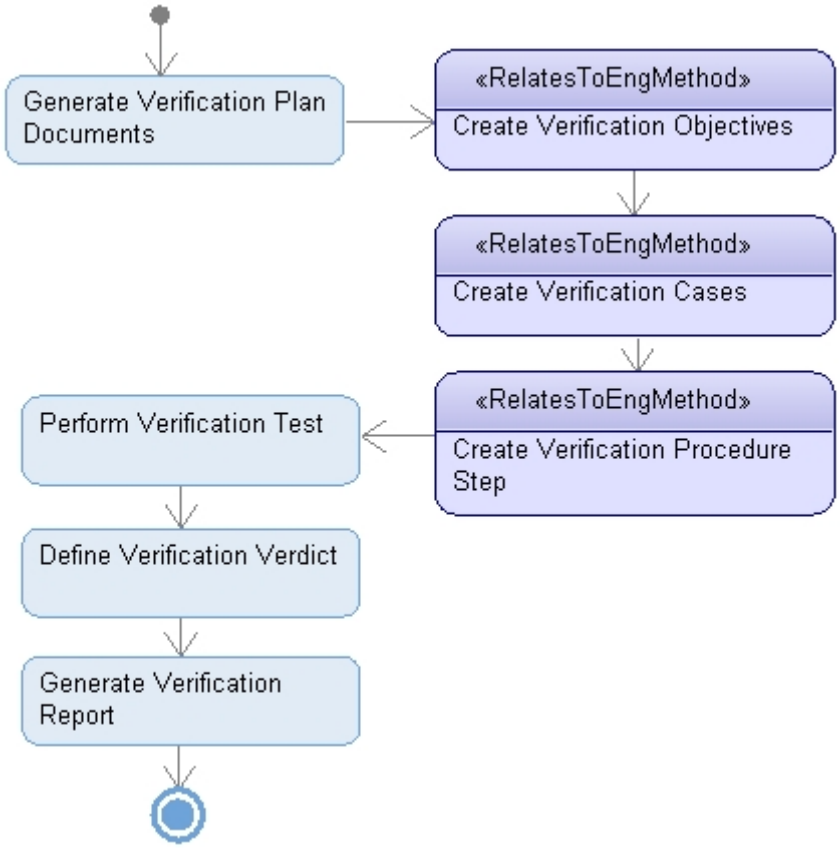

Figure 3-74 AD_PA206_03

Create Verification Cases (RelatesToEngMethod): Derive Verification Cases from Verification Objectives and allocate them to the related system requirements

Create Verification Objectives (RelatesToEngMethod): Derive Verification Objectives from System Requirement considering Design Aspects

Create Verification Procedure Step (RelatesToEngMethod): Decompose Verification Cases in Verification Steps and allocate the Verification Steps to the System Requirements. Verification steps contain stimuli and expected system behaviour.

Define Verification Verdict: Define the pass/fail criteria for the Verification cases respectively the Steps within the Verification Procedure.

Generate Verification Plan Documents: Generate the Verification Plan documents to provide information on the scope and test coverage.

Generate Verification Report: Generate a Report on the performed Verification tests, the test results and the pass/fail results.

Perform Verification Test: Perform the verification tests as defined in the verification procedure steps and record the test results.

3.5.3.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

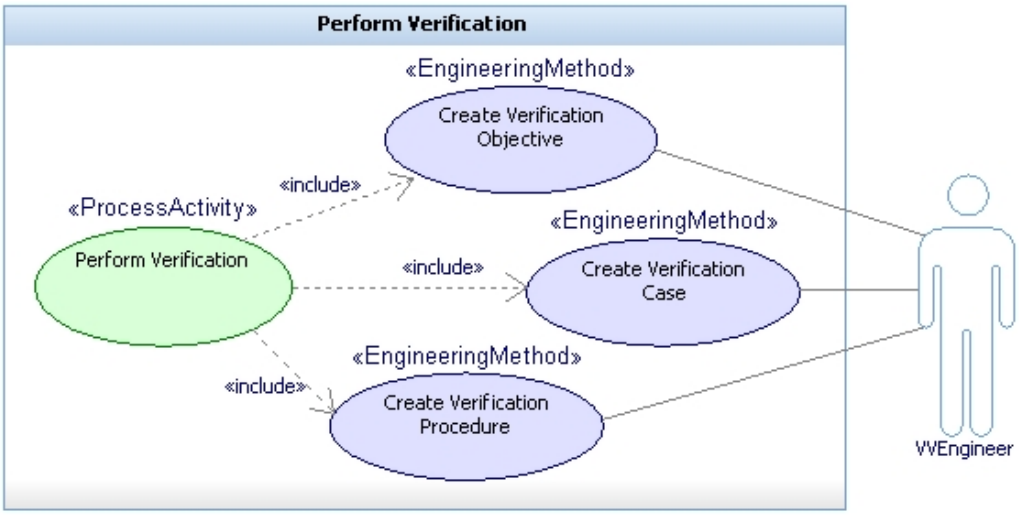

Figure 3-75 UCD_PerformVerification

3.5.3.2 EM206_03_01 - Create Verification Objective

ID: EM206_03_01

Derive Verification Objectives from System Requirement considering Design Aspects

EM Purpose: The verification objective and traceability to the selected requirement will be established.

EM Pre-Condition: The set of requirements to be verified is available.

EM Post-Condition: The verification objective of a selected requirement was created and a link to the selected requirement (link type = derived) was established.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Request the List of Requirements selected by the VVEngineer
02. display previous view on the requirements selected from the VVEngineer
03. Requirement for the definition of the verification objective selected by the VVEngineer
04a. Request for the detailed information of the selected requirement
04b. Request for design artefacts allocated to selected Requirement
04c. Create the empty verification objective for selected requirement
05. Generate Verification Objective based on requirement and design information
06. Establish "derive"-Link between Verification Objective and Requirement

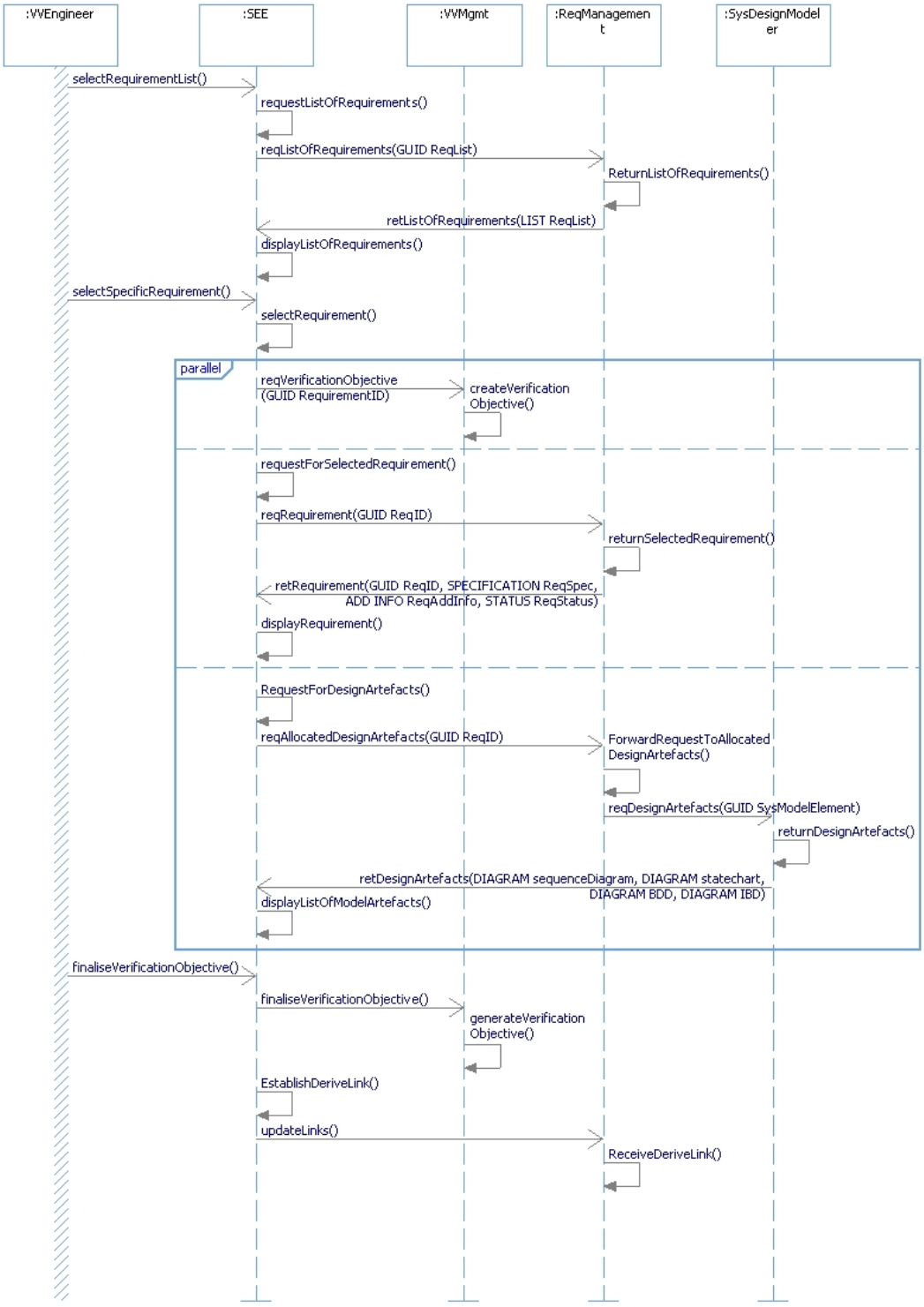

Figure 3-76 SD_EM206_03_01_Create Verification Objective

createVerificationObjective: Create the empty verification objective for selected requirement.

Relates to Engineering Activity: 04c. Create the empty verification objective for selected requirement

displayListOfModelArtefacts: Display the diagram containing the affected design artefacts

displayListOfRequirements: The VVMgmt display previous view on the requirements selected from the VVEngineer

Relates to Engineering Activity: 02. display previous view on the requirements selected from the VVEngineer

displayRequirement: Display requirement details on the selected requirement

EstablishDeriveLink: Establish "derive"-Link between Verification Objective and Requirement

Relates to Engineering Activity: 06. Establish "derive"-Link between Verification Objective and Requirement

ForwardRequestToAllocatedDesignArtefacts: Forward the request for the design artefacts to the SysDesignModeler according to the trace information

generateVerificationObjective: Generate Verification Objective based on requirement and design information.

Relates to Engineering Activity: 05. Generate Verification Objective based on requirement and design information

ReceiveDeriveLink: Accept "derive"-Link from Verification Objective to corresponding Requirement

RequestForDesignArtefacts: The VVMgmt tool forwards a request to the RequirementMgmt tool to provide the design artefacts allocated to the selected requirement

Relates to Engineering Activity: 04b. Request for design artefacts allocated to selected Requirement

requestForSelectedRequirement: The VVMgmt tool forward a request to the RequirementMgmt tool to provide the selected requirements

Relates to Engineering Activity: 04a. Request for the detailed information of the selected requirement

requestListOfRequirements: After the VVEngineer selected within a project a set of requirements for verification, e.g. a requirement module, the VVMgmt tool forward a request to the RequirementMgmt tool to provide the list of requirements

Relates to Engineering Activity: 01. Request the List of Requirements selected by the VVEngineer

returnDesignArtefacts: The SysDesignModeler return the diagrams containing the requested design artefacts (related to allocated requirement) to the VVMgmt.

ReturnListOfRequirements: The RequirementMgmt tool returns a preview on the requirements of the requested list

returnSelectedRequirement: The RequirementMgmt tool returns the requested requirements with required attributes.

selectRequirement: From the set of requirements a single requirement on which the verification objective shall be defined is selected by the VVEngineer.

Relates to Engineering Activity: 03. Requirement for the definition of the verification objective selected by the VVEngineer

3.5.3.3 EM206_03_02 - Create Verification Case

ID: EM206_03_02

Derive Verification Cases from Verification Objectives and allocate them to the related system requirements.

EM Purpose: The verification case and traceability to verification objective and requirement will be created.

EM Pre-Condition: The verification objectives are created and linked to the requirements (link type = derived).

EM Post-Condition: The verification case for a selected verification objective was created and links to the verification objectives (link type = derived) and to the allocated requirements (link type = verify) were established.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Create an empty verification case for the selected objective
02. Establish "derive"-Link between verification cases and objective
03. Generate the verification cases authorised by the VVEngineer
04. Get the requirements allocated to the selected verification objectives
05. Establish automatically the "verify"-Links between verification cases and requirements

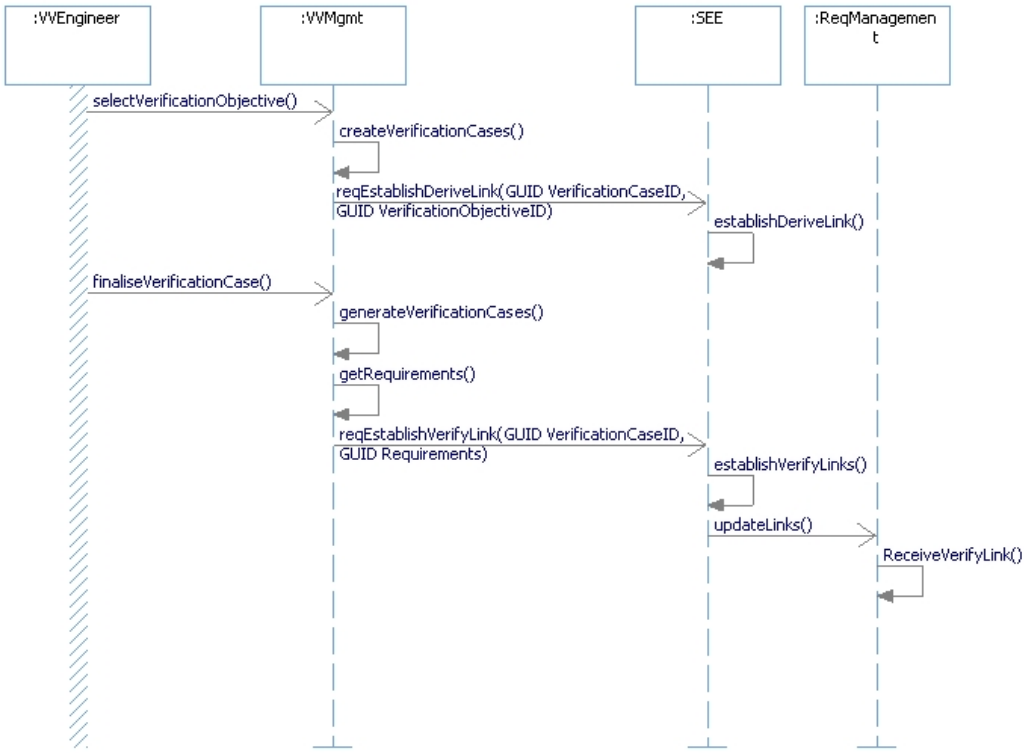

Figure 3-77 SD_EM206_03_02_Create Verification Case

createVerificationCases: After the selection of the verification objective by VVEngineer, an empty verification case for the selected objective is created

Relates to Engineering Activity: 01. Create an empty verification case for the selected objective

establishDeriveLink: Establish "derive"-Link between verification cases and objective.

Relates to Engineering Activity: 02. Establish "derive"-Link between verification cases and objective

establishVerifyLinks: Create automatically "verify"-Link between verification cases and requirements

Relates to Engineering Activity: 05. Create automatically "verify"-Link between verification cases and requirements

generateVerificationCases: Generate the verification cases authorised by the VVEngineer

Relates to Engineering Activity: 03. Generate the verification cases authorised by the VVEngineer

getRequirements: Get the requirements within the VVMgmt allocated to the selected verification cases

Relates to Engineering Activity: 04. Get the requirements within the VVMgmt allocated to the selected verification cases

ReceiveVerifyLink: Accept "verify"-Link from verification cases to corresponding requirements.

3.5.3.4 EM206_03_03 - Create Verification Procedure

ID: EM206_03_03

Decompose Verification Cases in Verification Procedure and allocate the Verification Procedure Step to the System Requirements

EM Purpose: The verification procedure steps and traceability to verification case and requirement will be created. The verify links of the procedure will replace the links of the verification case. The ones from the verification case will be deleted.

EM Pre-Condition: The verification cases are created and are linked to the verification objectives (link type = derive) and to the allocated requirements (link type = verify). The previous verify links from the selected verification case to the requirements were deleted.

EM Post-Condition: The verification procedure steps for a selected verification case was created and links to the verification cases (link type = decompose) and to the allocated requirements (link type = verify) were established.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Create an empty object for verification procedure steps
02. Create an "decompose"-Link between verification case and the verification procedure
03. Generate a verification procedure based on verification cases
04. Get the requirements within the VVMgmt allocated to the selected verification cases
05. Create automatically "verify"-Link between verification cases and requirements
06. Get the verification cases within the VVMgmt allocated to the selected to the verification procedure steps.
07. Remove the "verify"-Links between verification cases and requirements, because they are replaced by the verification procedure

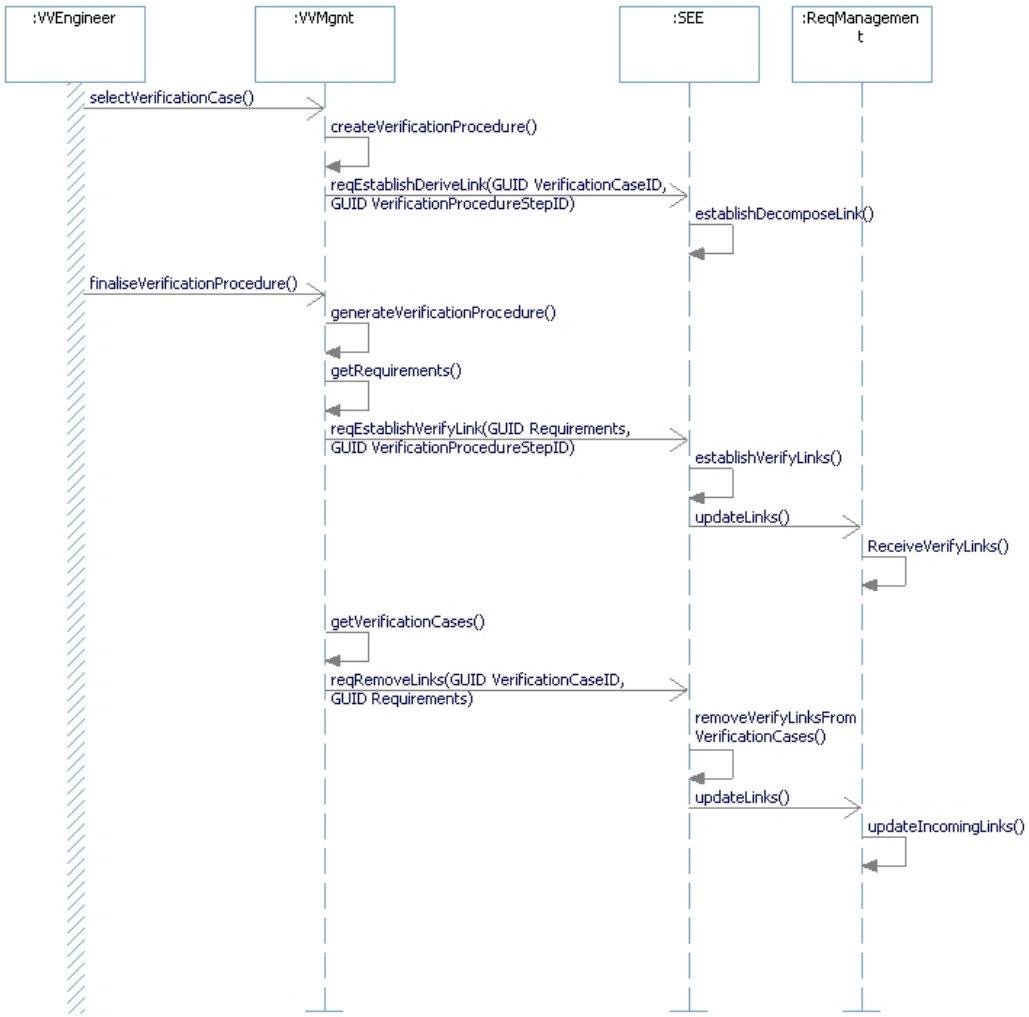

Figure 3-78 SD_EM206_03_03_Create Verification Procedure

createVerificationProcedure: After the VVEngineer selected the verification case to be described by a verification procedure an empty object for verification procedure steps will be created

Relates to Engineering Activity: 01. Create an empty object for verification procedure steps

establishDecomposeLink: Create an "decompose"-Link between verification case and the verification procedure

Relates to Engineering Activity: 02. Create an "decompose"-Link between verification case and the verification procedure

establishVerifyLinks: Create automatically "verify"-Link between verification cases and requirements

Relates to Engineering Activity: 05. Create automatically "verify"-Link between verification cases and requirements

generateVerificationProcedure: Generate a verification procedure based on verification cases

Relates to Engineering Activity: 03. Generate a verification procedure based on verification cases

getRequirements: Get the requirements within the VVMgmt allocated to the selected verification cases

Relates to Engineering Activity: 04. Get the requirements within the VVMgmt allocated to the selected verification cases

getVerificationCases: Get the verification cases within the VVMgmt allocated to the selected to the verification procedure steps.

Relates to Engineering Activity: 06. Get the verification cases within the VVMgmt allocated to the selected to the verification procedure steps.

ReceiveVerifyLinks: Accept "verify"-Links from verification cases to corresponding requirements

removeVerifyLinksFromVerificationCases: Remove the "verify"-Links between verification cases and requirements, because their are replaced by the verification procedure, which decomposes the verification case.

Relates to Engineering Activity: 07. Remove the "verify"-Links between verification cases and requirements, because they are replaced by the verification procedure

updateIncomingLinks: Update the incoming links to "remove" the deleted links.

3.5.4 PA206_04 - Analyse Impact of Change

ID: PA206_04

Related User Story: Project compliance monitoring based on advanced traceability

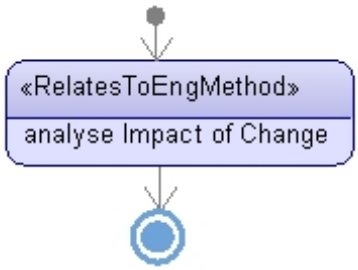

Figure 3-79 AD_PA206_04

analyse Impact of Change (RelatesToEngMethod): Perform impact analysis for an artefact to be changed based on the in-going links of an artefact. The Method is utilised in several other process activities to identify the artefacts (diagrams, requirements, models,...) that need to be changed due to a certain change.

3.5.4.1 Overview of Related Engineering Methods

The Process Activity relates to the Engineering Methods as depicted in the below diagram.

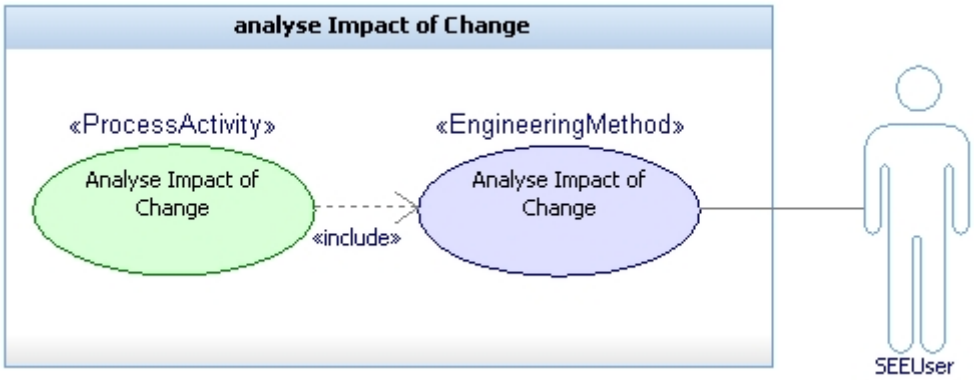

Figure 3-80 UCD_analyseImpactOfChange

3.5.4.2 EM206_04_01 - Analyse Impact of Change

ID: EM206_04_01

Perform impact analysis for an artefact to be changed based on the in-going links of an artefact.

EM Purpose: To get an overview for the impact of a change. All affected artefacts are reported.

EM Pre-Condition: Change proposal available and the traceability established during development.

EM Post-Condition: The change impact is illustrated in a report.

List of related Engineering Activities:

The Engineering Method is described by the following Engineering Activities:

01. Provide artefact IDs, of source artefacts linked to the requirement to be changed.
02. Illustrate the affected artefacts by an impact tree.
03. Provide the context information in which a selected artefacts interacts.
04. Display the context information requested for the specific artefact.

Figure 3-81 SD_EM206_04_01_Analyse Impact of Change

analyseChangeImpact: Analyse the impact of a change for a defined requirement.

displayArtefactContext: Display the context information requested for the defined artefact.

Relates to Engineering Activity: 04. Display the context information requested for the specific artefact.

illustrateImpactTree: Illustrate the affected artefacts by an impact tree. This can be done by tables, graphs or other demonstrative display options.

Relates to Engineering Activity: 02. Illustrate the affected artefacts by an impact tree.

provideAllLinkedArtefacts: Provide artefact IDs, of source artefacts linked to the requirement to be changed. Additionally follow refine and decomposition traces of requirement through all development levels and provide artefact IDs of affected low level artefacts.

Relates to Engineering Activity: 01. Provide artefact IDs, of source artefacts linked to the requirement to be changed.

provideContextOfArtefact: Provide the context information in which a selected artefacts interacts.

Relates to Engineering Activity: 03. Provide the context information in which a selected artefacts interacts.

providedDetailedInformation: On request of the System Engineer detailed information of an artefact are displayed.

3.6 Stakeholders & Roles

The following table provides an overview of the stakeholders and their roles with respect to the individual activities.

Stakeholders	**Role**
SystemsEngineer	The role of the System Engineer is to synthesize the system architecture and to design the systems details.
ConfigurationManager	The configuration manager is responsible for the configuration management. His goal is to have available at any time consistent baselines with a defined scope. The configuration and change manager may create change requests and is responsible to manage and monitor the change requests throughout its life cycle.
SEEAdmin	The administrator of the SE Environment.
SEEUser	The generic user of the SEE. He represents all the users i.e. use cases associated with the SEEUser are available to all users.
PLMSystem	The PLMSystem represents the long-term archive for the legal entity.
TechnicalApprover	The Technical Approver is the designated engineer who can approve decisions within the project.
ProjectManager	The PM interests in the SE environment are concerning the development progress and the achieved maturity.
HW_SW_Development	The implementation level (supplier, SW or HW development) receives data from the system level and provides the realized system element for integration and verification purposes.

RequirementsManager	The Requirements Manager defines, supports and monitors all activities related to Requirements Management for a specific project or program. This includes - tailoring and deployment of the requirements management process - developing a Requirements Management Plan, - provision of trainings, tools and infrastructure, - ensuring efficient development of requirements, - managing the configuration of requirements, - measuring the progress of the requirements management progress and the quality of requirements, - establishing and maintaining links with other processes (e.g. project management or risk management).
KnowledgeEngineer	The Knowledge Engineering is responsible for developing, analyzing and managing domain ontologies.
SafetyEngineer	The safety engineers analyses a proposed solution for the compliance with laws and regulations related to health and safety.
VVEngineer	The VVEngineers role is to perform the verification and validation and all the relating task.

Table 3-1 Stakeholders & Roles

4 Terms, Abbreviations and Definitions

4.1.1 Abbreviations

AD	Activity Diagram
API	Application Programming Interface
BDD	Block Definition Diagram
CA	Change Activity
CCB	Change Control Board
CI	Configuration Item
CM	Configuration Management
CO	[Dissemination Level]
ConOps	Concept Of Operations
COTS	Commercial Off The Self
CR	Change Request
CVL	Common Variability Language
DoDAF	Department of Defence Architecture Framework (US)
EM	Engineering Method
FDAL	Functional Development Assurance Level (analogue to ARP4754A)
FHA	Functional Hazard Analysis
FT	Fault Tree
FTA	Fault Tree Analysis
GUID	Graphical Unique Identifier
HW	Hardware
IBD	Internal Block Diagram
ICD	Interface Control Document
IDAL	Item Development Assurance Level (analogue to ARP4754A)
IOS	InterOperability Specification
IW	Innovation Works

KPI	Key Performance Indicator
LSI	Large Scale Integration
MBSE	Model-based Systems Engineering
MoDAF	Ministry of Defence Architecture Framework (UK)
MSE	Mission Support Equipment
PA	Process Activity
PLM	PLM
PM	Project Management
RM	Requirements Management
SC	State Chart
SD	Sequence Diagram
SE	Systems Engineering
SEE	Systems Engineering Environment
SW	Software
SysML	Systems Modelling Language
UAM	User Access Manager
UCD	Use Case Diagram
UPDM	Unified Profile for DoDAF/MODAF
US	User Story
VP	Variation Point
VV	Validation and Verification
WP	Work Package

Table 4-1 Abbreviations

4.1.2 Glossary

Artefact	An artefact is any type of object within the engineering development environment, that can be referenced to or is a configuration item of its own. Examples are requirements, models, model elements and files (.doc, .ppt,...).
Baseline	A Baseline is an approved and released set of artefacts having an association with the system or a dedicated configuration item. A baseline is managed by the configuration management and represents a reliable and consistent basis for subsequent design and development activities to which changes are addressed.

Baseline Set	A Baseline Set is an approved and released set of baselines of different configuration items across the various tools having an association with the system or a system element. The baseline set assures the consistency of the artefacts across the tools.
Baseline Set Definition	A Baseline Set Definition is a template for a baseline set containing a reference to all the configuration items which should be included in a baseline set. The Baseline Set Definition is a subset of all the configuration items available in the SEE environment serving a dedicated purpose. For example, including the allocated baseline for a next level team or a supplier. The baseline set definition contains baseline sets. Configuration items may be referenced by one or more baseline set definitions.
Boilerplates	Requirements boilerplates can be thought of as semi-complete requirements which are parameterized to suit a particular context. The parameters in a boilerplate generally refer to different attributes with respect to a given system (e.g. the system itself, stakeholders involved, and functions of the system, the objects and events involved in the system, performance characteristics, or units of measurement).
Concept of operation	The concept of operation is designed to give an overall picture of the operations using one or more specific systems, or set of related systems, in the operational environment from the users' and operators' perspective.
Configuration	The configuration is the configuration status of a single item, which status may change independently from other items. On the lowest levels we have: source code files having a dedicated version HW modules having a modification state documents having an issue Higher integrated configuration items and finally the system itself, are an arrangement of configuration items on a lower level, each having its own configuration state. The configuration of such an integrated item consists of a listing of all these lower level items and their configuration state. This configuration identifies also the selected options i.e. the variant.
Configuration Control Board	A board composed of technical and administrative representatives who recommend approval or disapproval of proposed engineering changes to, and proposed deviations from, an item's current approved configuration.
Configuration Item	A Configuration Item (CI) is any work product within the SEE designated for separate configuration management. A Configuration Item could be the complete system model or sub parts of it (packages), a requirements module, an analysis model, a simulation or mathematical model or a document. Single requirements and model elements are not treated as a configuration item as the granularity would be too fine and the consistency of a set of such elements is too complicated to ensure.

Configuration Management	Configuration Management (CM) is a process for establishing and maintaining consistency of a product's performance, functional and physical attributes with its requirements, design and operational information throughout its life. CM comprises following disciplines: Configuration Identification Configuration Control Configuration Status Accounting Configuration Verification and Audit
Configuration Status	Configuration Status is a report on the configuration baselines associated with each configuration item and all departures from the baseline, limitations and problems during design and production.
Controlled Vocabulary	Controlled vocabularies to organize knowledge for subsequent retrieval. They are used in indexing schemes, thesauri, taxonomies and other forms of knowledge organization schemes (source: Wikipedia). In CRYSTAL we use controlled vocabularies to build the ontology supporting requirements formalization.
Critical quality	Critical qualities relate to assurance, safety, security, and environment requirements.
Feature	Features are user-visible aspects or capabilities of a function of a product. Typically, a feature is one to two orders of magnitude coarser than a requirement. Features are a convenient way to characterize a function in terms that are understandable to various stakeholders. Features characteristics are "availability" in the products and "commonality". The AVAILABILITY of features and options shall be defined: - mandatory when available in all products. - optional features when selectable. - alternative features replacing other features. The COMMONALITY of features and options shall be defined: - common features available identical in all products. - variable features which might vary between the products. - customer specific features out of scope of the product family.
Ingoing Link	An ingoing link is, from the view of a link target, the link from a link source.
Language Defects	Language defects are syntactical issues in the formulation of natural language requirements. Examples are - omitting imperative shall (modal) - using passive voice instead of active voice
Link	A link is defined as the relation between a link source artefact (the origination of the link) and the link target artefact (the destination of the link)

Link Source	The origination of the link.
Link Target	The destination of the link.
Link type	The type of a link defines the characteristic of the relation. Links are grouped regarding their syntax and semantic, for example «refine», «verify» and «satisfy» are common link types. The SEE is capable to discriminate links according to their types, i.e. to filter links for one or more dedicated types.
Model Element	A model element is any object in a model that can be referenced. For example in a SysML model we have blocks, operations, transitions, states, events, data items, diagrams, views, packages.
Option	An Option defines exactly one possible resolution of a variation point.
Outgoing Link	An outgoing link is, from the view of a link source, the link to a link target.
Pattern	Requirements patterns use the same approach as Boilerplates - semi-complete templates are provided for the definition of Patterns which are parameterized to suit a particular context. In contrast to Boilerplates, these Patterns use a stronger formalism with fixed semantics (e.g. based on Linear Temporal Logic). As a result, Patterns are constraining users in a stronger way in writing requirements, but allow the application of numerous automatic analysis techniques (e.g. based on contracts and observers).
Physical Attributes	Physical characteristics (attributes) are quantitative and qualitative expressions of material features, such as composition, dimensions, finishes, form, fit, and their respective tolerances.
Product Family	A product family (also called product line) is a set of systems and products sharing a common, managed set of functions with several features, that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of building blocks in a prescribed way.

Product Variant Description	A Product Variant Description stores all information to create one specific product variant out of the product family assets. The product family feature model baseline is identified. Common features are identified. Selected variable feature are identified. The product family system requirements baseline is identified. Common requirements are identified. System requirements variation points are identified. Selected options and related system requirements are identified. The product family system model baseline is identified. Common product system model elements are identified. System model variation points and the selected product system model elements are identified. The product system model is identified. A Product Variant Description can be loaded to create a product variant.
Reference Technology Platform	A cross-domain standardised platform that provides meta-models, methods, and tools for safety-relevant hard real-time system development.
Requirement	A formalised statement identifying a capability, functionality, a physical characteristic or a quality that must be met or possessed by a system or system component to satisfy a contract, standard, a specification or other formally imposed documents. A requirement may be developed at any point in the product lifecycle by any number of stakeholders.
Requirement quality	Requirements shall fulfil quality characteristics such as CCC (complete, correct, and consistent) and SMART (specific, measurable, achievable, relevant, and traceable).
Requirements group	A set of requirements that serve the same role in the engineering process. Good examples are the requirements allocated to a particular System Element, the system requirements, or the assurance requirements featuring means of compliance and test objectives.
SEE Project File	The SEE Project File contains a list of references to ALL the working items for a functional SEE environment including requirements, system model, traceability information, workflow information, problem reports, change requests, analysis models, reports, documents.

SEE Workspace	The SEE workspace is the working environment and tool environment for the current user running on his local machine. The tools installed in this environment may differ depending on the role of the user. Safety engineers for example may have installed a tool to generate fault trees. The workspace need to be loaded with the SEE project file before the user can start working.
Stakeholder	An individual or organization having a right, share, claim, or interest in a system or its characteristics that meet their needs and expectations. NOTE Stakeholders include, but are not limited to end users, end user organizations, supporters, developers, producers, trainers, maintainers, disposers, acquirers, customers, operators, supplier organizations, creditors, and regulatory bodies.
Stakeholder Needs	Expectations stakeholders have about the characteristics of the system that may not necessarily be clear, consistent or even achievable.
Suspect Link	Linked objects are marked as having suspect links if the object that they link to has changed.
System Engineering Environment	The System Engineering Environment (SEE) is the tool environment for the system engineer including tools for different purposes and concepts. The tools are collaborating to automate tasks within the environment and to establish common tool independent methods. A focus is on Model Based System Engineering with a system design model as the common link between all the different tasks and models.
Systems Engineering	An interdisciplinary approach and means to enable the realization of successful systems. This approach starts with the definition of stakeholder needs, the identification of product functionality and the intended validation very early in the lifecycle. Systems engineering considers both the business and the technical needs of all stakeholders with the goal of providing a quality product that meets the user needs.
Traceability	The ability to identify the relationship between various artefacts of the development process, i.e. the lineage of requirements, the relationship between a design decision and the affected requirements and design features, the assignments of requirements to design features, the relationship of test results to the original source of requirement. Bi-directional traceability is required to permit top-down impact analysis and down-top traceability analysis.
Transformation Protocol	A transformation protocol documents the variability resolution during the creation of a product variant: - common and selected features - system requirements variation points and selected options - identified system requirements - system model variation points and selected options - identified system model elements For each creation of a product variant a transformation protocol shall be created.

User Story	A User Story describes a typical action pattern or work flow within an industrial domain. The user stories are used to describe general processes that are too high level to derive development requirements directly out of it.
Validation	The purpose of the validation process is to provide objective evidence that the services provided by a system when in use comply with stakeholders' requirements, achieving its intended use in its intended operational environment, i.e. assuring the right system will be designed and built. This process performs a comparative assessment and confirms that the stakeholders' requirements are correct, complete and consistent (CCC). Where variances are identified, these are recorded and guide corrective actions.
Variant	A variant selects options of a variation point (e.g. a red car, a green car) or a product with several options selected for its variation points making it different to other products based on the same specifications.
Variation Point	A variation point is a representation of a subject or attribute that can vary (e.g. colour of a car). Options are linked to the variation point and show the range of the variability (e.g. colour red, green blue; for some reasons other colours are not available). A Variant identifies a single option of a variation point (e.g. a red car, a green car). The specification of Variation Point definitions shall include: - the description WHAT shall vary, - the identification of the possible options in order to define range of variation, - definition of the binding time when the options are implemented into the product (compile/link time, integration time, installation time, operation time). - the rationale why this variation is required or how it will pay off - the stakeholder requesting for the variation - the visibility of the variation point (internal or external) shall be defined.
Verification	The purpose of the verification process is to confirm that the specified requirements are fulfilled by the system, i.e. assuring the system has been designed and built right. This process provides the information required to effect the remedial actions that correct non-conformances in the realized system or the processes that act on it.

Table 4-2 Terms

5 References

If no other remarks are given the last valid issue of the mentioned document applies.

ARP4754A	Certification consideration for highly integrated or complex aircraft systems
CRYSTAL DOW	Critical System Engineering Acceleration, Annex I - "Description of Work", Grant agreement no: 332830
IBM Harmony SE	Hans-Peter Hoffmann, IBM Rational Harmony for Systems Engineering, Deskbook, Release 3.1.2
ISO IEC 15288 2008	Systems and Software Engineering - System Lifecycle Processes
ISO IEC 29148 2011	Systems and Software Engineering - Life Cycle Processes - Requirements Engineering

6 Annex I: Technology Baseline & Progress Beyond

The following table describes the technology base line and progress beyond in regard to the user stories.

ID	Name	Technology Base Line	Progress Beyond
US202	Safety Analysis	Safety Engineering is today performed segregated from the systems engineering activities. This means that the system architectural and functional relevant information has to be extracted manually from system engineering artefacts and be entered into the safety analysis tools for evaluation. The resulting reliability figures than are entered into the SE domain again manually as safety requirements. Consistency of calculated and specified reliability figures has to be performed manually.	The bricks (FaultTree+) supporting the use case allow the automatic extraction of fault tree relevant architecture, function and failure case information from a given system model (SysML) to build a library of fault tree elements as a basis for fault tree definition. Failure case related severity classifications stored in the SysML model are transferred into required reliability figures to be optionally assigned to the fault tree elements. An automated check allows the comparison of system and fault tree model information for coherence.
US203	Variability Management	Feature based variability management has been partially established on system requirements level. The system architecture variability has been modelled in SysML.	Functions, Features and feature options, relations and constraints shall be described in a product family feature model. Variability, relations and constraints within system requirements and system architecture model shall be described for product families. Resolving the variability by using the SEE the subset of system requirements and a product system model for a product variant shall be provided.
US204	Ontology-based Requirements Engineering	Requirements are analysed manually using checklists and peer reviews. Some proprietary add-on tools have been developed which allow to calculate basic quality metrics.	Requirements quality analysis is performed automatically. Comprehensive quality metrics are calculated. Findings provide recommendations guiding the improvement of requirements.
US205	Process Automation, Guidance and Monitoring	Change Control: Change requests are managed in a separate tool. Often integration for SW artefacts and source code is considered but not all SE artefacts. Issue tracking (problems and changes) are file based and are difficult to adapt to tools with a data base or binary data. Only textual	Change Control: Change requests are managed at a single place for all the tools. The user has the possibility to add links to any of the artefacts in the SEE environment. The links allow easy navigating to the affected artefact. The way tools data is organized is transparent to the change management and other functions i.e. it does not

ID	Name	Technology Base Line	Progress Beyond
		references to affected artefacts are possible. Strict change control for SE tools and databases is not possible. Implementation of change control for each tool is expensive. Configuration Management: Today Configuration management is performed separate for each tool in a different way. Tools do not support baselining in a same way. There are many tools not supporting baselining. There the user is forced to collect all the files and to put them under configuration control bearing the risk to miss necessary artefacts. Implementation of version management and baselining is not possible for a lot of tools. Expensive to implement for each tool. Consistency of the data needs to be maintained manually. Process Automation: Only large (and expensive) tools suits support automation. Several tools do not support automation. Report generation using data across several tools is not possible. Or each tool has its own API. Tools are not much integrated. Examples: - Tool Inventory needs to be generated manually. - Revision History needs to be generated for each tool in a different way. User and User Access Management: Many tools do not support user and user access management. Tools supporting such features each having its own management and	matter if data is stored in local file system or in a data base hosted by a server. Configuration Management: All the SEE tools support baselining and import / export in a same way controlled by the SEE from a single point. The SEE provides common version management and baselining features and provides access to these functionalities for each of the SEE registered tools allowing them to participate. The SEE environment support maintenance of data consistency. Process Automation: Tools provide a common API i.e. one purpose same function. The SEE provides for common tasks according scripts. User and User Access Management: All the tools in the SEE are participating in the user and user access management. There is one instance to manage the users and the group rights down to level of the configuration items.

ID	Name	Technology Base Line	Progress Beyond
		user configuration. The administrator needs to maintain several data sets for a single user. If tools are supporting user and user access management, users needs to authenticate them at every tool. Single sign on is provided on top by third parties.	
US206	Project compliance monitoring based on advanced traceability	Traceability of artefacts today is only available, if at all, within an engineering tool. To link artefacts over tool chains is not yet possible. Some engineering tools are able to import a copy of the artefacts to be linked. The imported artefacts could then be linked and are traceable within this tool. For example the engineering tools: DOORS and Rhapsody. Today the requirements from DOORS can be imported by an Rhapsody Tool Add-IN (Gateway) into the Rhapsody Model. Within the model, the imported requirements can be linked and are traceable. To provide this information back to DOORS, the Rhapsody Model Elements are exported to DOORS. At least the links within DOORS are then automatically established. A lot of tools within the system engineering environment do not support traceability at all. Some do not provide the capability for linking at all, other in fact provide the capability of linking but it is not guaranteed that the link is also traceable.	The bricks supporting the use case allow the linking of different artefacts over various tools integrated in the SEE. Therefore several link types are provided but additionally user required link types can be defined. The bricks support the bi-directional access to the links within the SEE and readable access to their source and target artefacts and can therefore provide the capability to use this information for change impact and coverage analyses, report generation, requirements management, rationale management, verification and validation tasks and so on.. The bricks assist the SEEUser by creating and maintaining the links automatically as often as possible.

Table 6-1Technology Baseline & Progress Beyond

7 Annex II: Detailed Descriptions of the Engineering Methods

7.1 PA202_01 - Perform Functional Safety Analysis

7.1.1 EM202_01_01 Transfer Model Data for Fault Tree Analysis

Engineering Method: US202_Transfer Model Data for Fault Tree Analysis_EM202_01_01					
Purpose: Logical blocks with their primitive operations and/or the physical blocks from the system model are made available in the fault tree tool in order to create representations for the fault tree anaylsis.					
Comments: Reliability figures are stored within the system model. Rationale: The system model is part of the specification as such. The fault tree as an analysis model is not (it is the justification for reliability figures required).					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Logical architecture with primitive operations and/or physical architecture are available in the system model.		01. Extract System Use Cases from model 02. Select Use Case and type for FTA 03. Extract physical architecture data (physical FTA only) 04. Extract logical architecture data 05. Generate basic elements library 06. Establish traces to originating elements		Model elements transformed into a set of artefacts (basic elements) as input for fault tree analysis.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	SystemArchitecture	Name:	-	Name:	-
Generic Type:	SysMLModel	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	SysUseCase	Name:	-	Name:	-
Generic Type:	SysML_UseCas	Type:	-	Generic Type:	-

	e				
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	LogicalBlock	Name:	-	Name:	-
Generic Type:	SysML_Block	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	PhysicalBlock	Name:	-	Name:	-
Generic Type:	SysML_Block	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	SystemFunction	Name:	-	Name:	-
Generic Type:	SysML_Primitive Operation	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Fault Tree Elements Library
Generic Type:	-	Type:	-	Generic Type:	FaultTree_Elements
Required Properties:	-	Properties:	-	Provided Properties:	

Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	SystemSafetyTrace
Generic Type:	-	Type:	-	Generic Type:	Link
Required Properties:	-	Properties:	-	Provided Properties:	ArtefactStatus LinkSource LinkTarget
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: LinkSource = SystemModelElement LinkTarget = FaultTreeElement ArtefactStatus = Validated	

Table 7-1 EM202_01_01 Transfer Model Data for Fault Tree Analysis

7.1.2 EM202_01_02 Check System Model and Fault Tree Consistency

Engineering Method: US202_Check System Model and Fault Tree Consistency_EM202_01_02					
Purpose: Check that the functions, related failure cases and severity classification is consistent to the fault tree elements and their related reliability figures.					
Comments: none					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
FTA performed		01. Set System Model Scope 02. Retrieve FTA Data 03. Correlate System Model Elements With FTA Elements 04. Compare Reliability Figures 05. Generate Safety Coherence Report		Traceability between model and fault tree elements established	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	SystemArchitecture	Name:	-	Name:	-
Generic Type:	SysMLModel	Type:	-	Generic Type:	-

Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	SystemFTA	Name:	-	Name:	-
Generic Type:	FaultTree	Type:	-	Generic Type:	-
Required Properties:	Content	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints: Content = Fault Tree (Validated)		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	ConsistencyReport
Generic Type:	-	Type:	-	Generic Type:	Report
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	SysUseCase	Name:	-	Name:	-
Generic Type:	SysML_UseCase	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	PhysicalBlock	Name:	-	Name:	-
Generic Type:	SysML_Block	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Name:	LogicalBlock	Name:	-	Name:	-
Generic Type:	SysML_Block	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	SystemFunction	Name:	-	Name:	-
Generic Type:	SysML_Primitive Operation	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	SystemSafetyTrace	Name:	-	Name:	-
Generic Type:	Link	Type:	-	Generic Type:	-
Required Properties:	ArtefactStatus LinkSource LinkTarget	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints: LinkSource = SystemModelElement LinkTarget = FaultTreeElement ArtefactStatus = Validated		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-2 EM202_01_02 Check System Model and Fault Tree Consistency

7.2 PA203_01 - Product Family Domain Engineering

7.2.1 EM203_01_01 Develop Domain System Requirements

Engineering Method: US203_Develop Domain System Requirements_EM203_01_01					
Purpose: The variability of the product family system requirements shall be managed consistently to variable functions, features.					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
System Requirements defined for a product family. Optional requirements are identified. Variable system requirements identified by variation points. Functions and features modeled in a feature model. variable functions and features identified.		01. create product family system requirements 02. define system requirements variation points 03. request functions and features of the product family 04. extract functions features of the product family 05. provide functions and features requirements data of the product family 06. display functions, features and allocated requirements 07. allocate system requirements to common features or feature options 08. link requirements variation points to features 09. check linkage between requirements and features		System Requirements are linked to features. System requirements variation points are defined, options are specified. Relations between features and variation points are described.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Feature List for the Product Family	Name:	-	Name:	-
Generic Type:	FeatureList	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Function list of the product family	Name:	-	Name:	-
Generic Type:	FunctionsList	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Product Family requirements Module including resolution conditions
Generic Type:	-	Type:	-	Generic Type:	RequirementsModule
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	Product Familily Requirements Module	Name:	-	Name:	-
Generic Type:	RequirementsModule	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-3 EM203_01_01 Develop Domain System Requirements

7.2.2 EM203_01_02 Develop Reference Architecture

Engineering Method: US203_Develop Reference Architecture_EM203_01_02					
Purpose: The variability of the product family system model shall be managed consistently to variable functions and features.					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
The system architecture is developed. Functions and features variability is defined.		01. define logical system view 02. validate logical system view 03. merge system states 04. create reference architecture 05. identify variable system model elements 06. request product family functions and features 07. extract product family functions and features 08. provide product family functions and features 09. link system model elements to features or feature options 10. complete resolution conditions of system model variation points 11. perform model check		The system model artefacts are linked to functions and features of the feature model. Resolution rules have been created in the system architecture model.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Feature List of the Product Family	Name:	-	Name:	-
Generic Type:	FeatureList	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Function List of	Name:	-	Name:	-

	the Product Family				
Generic Type:	FunctionsList	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	System Model including resolution conditions
Generic Type:	-	Type:	-	Generic Type:	SysMLModel
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Table 7-4 EM203_01_02 Develop Reference Architecture

7.3 PA203_02 - Product Realization

7.3.1 EM203_02_01 Create Product System Requirements

Engineering Method: US203_Create Product System Requirements_EM203_02_01					
Purpose: The system requirements for a product variant shall be created consistently to functions, features.					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Variation points and resolution rules are defined for functions and feature, system requirements of the product family. Functions, Features of the product variant are identified in a product variant description.		01. require list of product variant descriptions 02. create list of product variant descriptions 03. provide list of product variant descriptions 04. select a product variant description 05. provide a product variant description 06. identify system requirements 07. identify and resolve system requirements variation points 08. display the system requirements of the product variant 09. review the system requirements of the product variant 10. change feature selection for the product variant 11. store the requirements resolution 12. enhance product variant description with the requirements resolution 13. store the transformation protocol of the system requirements		System requirements variation points are resolved and system requirements are available for a product variant. A transformation protocol documents the decisions leading to the product system requirements. The product variant description is enhanced by the identified product system requirements.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	List of Product Variant	Name:	-	Name:	-

	Descriptions				
Generic Type:	ProductVariantDescriptionsList	Type:	-	Generic Type:	-
Required Properties:	ProductListType	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Product Variant Description	Name:	-	Name:	-
Generic Type:	ProductVariantDescription	Type:	-	Generic Type:	-
Required Properties:	ProductFunction ProductFeature ProductFunction ProductFeature ProductRequirement	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Product Requirements Transformation Protocol
Generic Type:	-	Type:	-	Generic Type:	RequirementsTransformationProtocol
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	Product Requirements List
Generic Type:	-	Type:	-	Generic Type:	RequirementList

Required Properties:	-	Properties:	-	Provided Properties:	RequirementList Type
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: RequirementListType = ProductVariant	
Name:	-	Name:	-	Name:	Product Variant Description including Product Requirements
Generic Type:	-	Type:	-	Generic Type:	ProductVariantDescription
Required Properties:	-	Properties:	-	Provided Properties:	ProductFunction ProductFeature ProductFunction ProductFeature ProductRequirement
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Table 7-5 EM203_02_01 Create Product System Requirements

7.3.2 EM203_02_02 Create Product System Architecture

Engineering Method: US203_Create Product System Architecture_EM203_02_02		
Purpose: The system model for a product variant shall be created consistently to functions, features and system requirements.		
Pre-Condition	**Engineering Activity**	**Post-Condition**
Variation points and resolution rules are defined in the reference architecture of the product family. Functions, Features and System Requirements are identified in a product variant description.	01. require list of product variant descriptions 02. create list of product variant descriptions 03. provide list of product variant descriptions 04. select a product variant description 05. provide the product variant description	System model variation points are resolved and a system model is created for a product variant. A transformation protocol documents the decisions leading to this product system model. The product variant description is enhanced by the identified product system model elements.

		06. identify the system model elements for the product variant 07. resolve system model variation points 08. display product system model 09. review system model for the product variant 10. change product feature selection 11. store resolution of system model variation points 12. enhance product variant description by the product system model elements 13. store transformation protocol of system model			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	List of Product Variant Descriptions	Name:	-	Name:	-
Generic Type:	ProductVariantDescriptionsList	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	
Generic Type:	-	Type:	-	Generic Type:	
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	Product Variant Description

					including System Model Elements
Generic Type:	-	Type:	-	Generic Type:	ProductVariantDescription
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	Product Variant Description	Name:	-	Name:	-
Generic Type:	ProductVariantDescription	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Transformation Protocol of Product System Model Elements
Generic Type:	-	Type:	-	Generic Type:	SystemModelTransformationProtocol
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Table 7-6 EM203_02_02 Create Product System Architecture

7.4 PA204_01 - Perform Requirements Engineering

7.4.1 EM204_01_01 Define Requirements

Engineering Method: US204_Define Requirements_EM204_01_01					
Purpose: Support the requirements author during writing by requirements quality evaluation.					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Requirement is stored in the requirements management database. Initially, the requirement may be empty.		01. Select a requirement for editing 02. Retrieve the selected requirement from the requirements management database 03. Display the selected requirement 04. Retrieve the boilerplates of the boilerplate group assigned to the current requirements module 05. Match the current requirement with the boilerplates in order to find the best fitting boilerplates 06. Propose terms of the controlled vocabulary fitting with the matched boilerplates 07. Analyze requirements quality and create findings in order to guide improvement 08. Display requirements quality findings 09. Improve the requirement based on requirements quality findings 10. Update the requirement in the requirements management database.		Improved requirement is stored in the requirements management database.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Ontology	Name:	-	Name:	-
Generic Type:	DomainOntology	Type:	-	Generic Type:	-
Required		Properties:	-	Provided	-

Properties:				Properties:	
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Evaluation Results
Generic Type:	-	Type:	-	Generic Type:	QualityEvaluationResult
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	Boilerplates	Name:	-	Name:	-
Generic Type:	Boilerplate	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Requirements	Name:	-	Name:	-
Generic Type:	Requirement	Type:	-	Generic Type:	-
Required Properties:	ObjectText ID ObjectText RequirementLevel RequirementType AdditionalInformation Rationale Assumption IsDerived Source	Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Links from Requirement to Requirement	Name:	-	Name:	-
Generic Type:	satisfy	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints: LinkSource = Requirement LinkTarget = Requirement ArtefactStatus = valid		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Improved Requirements
Generic Type:	-	Type:	-	Generic Type:	Requirement
Required Properties:	-	Properties:	-	Provided Properties:	ObjectText ID ObjectText RequirementLevel RequirementType AdditionalInformation Rationale Assumption IsDerived Source
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	Links from Evaluation result to Requirement
Generic Type:	-	Type:	-	Generic Type:	refer
Required	-	Properties:	-	Provided	

Properties:				Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: LinkSource = Evaluation Result LinkTarget = Requirement ArtefactStatus = valid	

Table 7-7 EM204_01_01 Define Requirements

7.4.2 EM204_01_02 Search Requirements

Engineering Method: US204_Search Requirements_EM204_01_02					
Purpose: Support the requirements engineering in reusing previously defined requirements.					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Requirements are stored in the requirements management database.		01. Formalize natural language statement into knowledge structures that support reasoning and retrieval 02. Store knowledge structure 03. Define search criterion. 04. Retrieve knowledge structures including references to requirements matching the search criterion 05. Retrieve referenced requirements from requirements management database 06. Select requirements for analysis		Formalized requirements are stored in knowledge base for search and retrieval.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Ontology	Name:	-	Name:	-
Generic Type:	DomainOntology	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Boilerplates	Name:	-	Name:	-
Generic Type:	Boilerplate	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Requirements
Generic Type:	-	Type:	-	Generic Type:	Requirement
Required Properties:	-	Properties:	-	Provided Properties:	ID ObjectText RequirementLevel RequirementType AdditionalInformation Rationale Assumption IsDerived Source
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Table 7-8 EM204_01_02 Search Requirements

7.4.3 EM204_01_03 Analyze Requirements Quality

Engineering Method: US204_Analyze Requirements Quality_EM204_01_03					
Purpose: Improve requirements quality by requirements quality evaluation.					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Requirements are stored in the requirements management database.		01. Define quality metrics 02. Select requirements module 03. Assign quality metrics to requirements module 04. Analyze requirements module 05. Retrieve requirements for analysis 06. Select requirements for analysis 07. Analyze requirements quality and create findings in order to guide improvement 08. Create requirements quality report 09. Improve requirements based on requirements quality findings 10. Update requirements in the requirements management database		Improved requirements are stored in the requirements management database including links to quality evaluation results.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Ontology	Name:	-	Name:	-
Generic Type:	DomainOntology	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Requirements Module	Name:	-	Name:	-

Generic Type:	RequirementsModule	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Evaluation Results
Generic Type:	-	Type:	-	Generic Type:	QualityEvaluationResult
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	Quality Reports
Generic Type:	-	Type:	-	Generic Type:	RequirementsQualityReport
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	Improved Requirements
Generic Type:	-	Type:	-	Generic Type:	Requirement
Required Properties:	-	Properties:	-	Provided Properties:	ObjectText ID ObjectText RequirementLevel RequirementType AdditionalInformation Rationale

					Assumption IsDerived Source
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	Boilerplates	Name:	-	Name:	-
Generic Type:	Boilerplate	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Requirements	Name:	-	Name:	-
Generic Type:	Requirement	Type:	-	Generic Type:	-
Required Properties:	ObjectText ID ObjectText RequirementLevel RequirementType AdditionalInformation Rationale Assumption IsDerived Source	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Links from Requirement to Requirement	Name:	-	Name:	-
Generic Type:	satisfy	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints: LinkSource = Requirement LinkTarget = Requirement ArtefactStatus = valid		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Links from Evaluation result to Requirement
Generic Type:	-	Type:	-	Generic Type:	refer
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: LinkSource = Evaluation Result LinkTarget = Requirement ArtefactStatus = valid	

Table 7-9 EM204_01_03 Analyze Requirements Quality

7.5 PA205_01 - Configuration Management

7.5.1 EM205_01_01 Create Baseline Set Definition

Engineering Method: US205_Create Baseline Set Definition_EM205_01_01					
Purpose: Generate a baseline set definition as a template for the baseline sets.					
Comments: -					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Project file and configuration items are available.		01. Create the Baseline Set Definition List and request a list of config items from each tool. 02. The Tool provides the list of all its config items. 03. For each tool select the config items from the list provided. 04. Store relevant information in the baseline set definition list for each configuration item. 05. For each configuration item the dependencies are requested from the hosting tool. 06. The tool provides for the given configuration item all dependencies to other configuration items. 07. The SEE removes from the received list all configuration items already considered in the definition. 08. The user selects from the configuration items not yet considered those he wants to add to the baseline set definition.		Baseline Set Definition generated containing all the configuration items of interest for a dedicated purpose, scope or delivery.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:		Name:	-	Name:	-
Generic Type:		Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	
Generic Type:	-	Type:	-	Generic Type:	
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Table 7-10 EM205_01_01 Create Baseline Set Definition

7.5.2 EM205_01_02 Branch Baseline

Engineering Method: US205_Branch Baseline_EM205_01_02					
Purpose: To generate a new branch or working stream.					
Comments: -					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
A trunk or a parent branch is available with baseline sets established.		01. The configuration manager requests to generate a branch from any baseline set. 02. At the SEE the user selects one baseline set from the available baseline sets. For each configuration item the SEE asks the tool if it supports the history feature. 03. Show the status of the complete operation.		A new branch is created.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	ProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Definition	Name:	-	Name:	-
Generic Type:	BaselineSetDefinition	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	trunk	Name:	-	Name:	-
Generic Type:	BaselineSet	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	NewBranch
Generic Type:	-	Type:	-	Generic Type:	BaselineSet
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: Status = Created	

Table 7-11 EM205_01_02 Branch Baseline

7.5.3 EM205_01_03 List Change History

Engineering Method: US205_List Change History_EM205_01_03		
Purpose: To generate a delta history report.		
Comments: -		
Pre-Condition	**Engineering Activity**	**Post-Condition**
Baseline set definitions are available and at least two baseline sets are established.	01. The user requests to generate a history report. 02. At the SEE the user selects one	A report is generated listing all the changes and modifications applied for the configuration items from one baseline set to a following baseline

	baseline set from the available baseline sets. For each configuration item the SEE asks the tool if it supports the history feature. 03. The tool provides the information if it supports the history feature in general and for the current data base. 04. The SEE checks the history status. 05. If history is not supported, the SEE skips the configuration item and adds an appropriate information in the report. 06. The SEE requests for the given configuration item the delta information to its previous baseline. 07. The tool provides the delta information for the given baseline of the configuration item. 08. The SEE adds the delta information to the report. 09. The user requests a full history and selects the baseline set definition. For each configuration item in all the baseline sets, the SEE asks the tool if it supports the history feature. 10. The tool provides the information if it supports the history feature in general and for the current data base. 11. The SEE checks the history status. 12. If history is not supported, the SEE skips the configuration item and adds an appropriate information in the report. 13. The SEE requests for the given configuration item the delta information to its previous baseline. 14. The tool provides the delta information for the given baseline of the configuration item.	set.

		15. The SEE adds the delta information to the report.			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Definition	Name:	-	Name:	-
Generic Type:	BaselineSetDefinition	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Base	Name:	-	Name:	-
Generic Type:	BaselineSet	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Changed	Name:	-	Name:	-
Generic Type:	BaselineSet	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Deltas
Generic Type:	-	Type:	-	Generic Type:	Report
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Name:	ProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-12 EM205_01_03 List Change History

7.5.4 EM205_01_04 Merge Baselines

Engineering Method: US205_Merge Baselines_EM205_01_04		
Purpose: To integrate changes of different branches (work streams) into an other branch or the trunk.		
Comments: -		
Pre-Condition	**Engineering Activity**	**Post-Condition**
Changes made to a branch	01. The user requests the comparison of two baselines. 02. The user selects an available baseline set definition. 03. From the list of available baseline sets the user selects the two baseline sets to compare. 04. The SEE lists all the configuration items with the version number for both baseline sets highlighting different version numbers. 05. The configuration manager checks the changes made 06. The configuration manager has decided to replace a configuration item completely. The tool is requested to replace the configuration item. 07. The tool replaces the identified configuration item in the parent branch or trunk with the modified item from the branch. The tool provides an operation status, 08. The SEE collects the operation status and displays the results. 09. The user selects a	Changes made in a branch (work stream) are integrated into the parent branch or the trunk.

		configuration item and requests a detailed comparison. 11. The tool loads the two baselines of the given configuration item. 12. The tool visualizes the detailed changes of the two baselines. 13. The configuration manager accepts those changes to be incorporated from the branch into its parent branch or the trunk. 14. The tool incorporates the selected modification. 15. The user continues until all changes from the baseline are incorporated. 16. The configuration manager may decide to cancel the branch or to continue the branch. In case the branch is cancelled, working on that branch is no more possible.			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	ProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Definition	Name:	-	Name:	-
Generic Type:	BaselineSetDefinition	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	trunk	Name:	-	Name:	-
Generic Type:	BaselineSet	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Branch	Name:	-	Name:	-
Generic Type:	BaselineSet	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints: Status = Available		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	BranchDeleted
Generic Type:	-	Type:	-	Generic Type:	BaselineSet
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: Status = Deleted	

Table 7-13 EM205_01_04 Merge Baselines

7.5.5 EM205_01_05 Export Import Baseline Set

Engineering Method: US205_Export Import Baseline Set_EM205_01_05		
Purpose: To create a copy of the SEE data in the local file system for archiving and restoring the workspace from an archive.		
Comments: -		
Pre-Condition	**Engineering Activity**	**Post-Condition**
Project file and baseline sets are available.	01. The SEEUser selects an available baseline set. 02. The SEE activates the baseline	Workspace and project files restored from previously exported archive.

	set. See "loadArtefact". 03. The CM starts export by selecting a baseline set. 04. The SEE creates a new archive. 05. The SEE stores relevant information from the baseline set in the archive. 06. For each configuration item the SEE requests the associated tool to export the baseline as defined by the baseline set. 07. The tool exports the requested baseline of the given configuration item into the archive. 07. The tool exports the requested baseline of the given configuration item into the archive. 08. The SEE collects and combines the results of each operation. 09. The CM starts the import of a baseline and selects the source archive. 10. The SEE loads the information on the baseline set. 11. The SEE requests each tool to create a new workspace. 12. Each tool creates a new workspace e.g. an empty database. 13. The SEE collects and combines the results of each operation. 14. For each config item in the baseline set, the SEE requests the associated tool to import the configuration item from the source archive. 15. Each tool imports the requested baseline of the given configuration item from the source archive. 16. The SEE collects and combines the results of each operation.	
Notes: -	Notes: -	Notes: -
Artefacts Required as inputs of	**Artefacts used internally within**	**Artefacts Provided as outputs of**

the Activities		**the Activities (optional)**		**the Activities**	
Name:	ProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Definition	Name:	-	Name:	-
Generic Type:	BaselineSetDefinition	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Baseline Set to be exported	Name:	-	Name:	-
Generic Type:	BaselineSet	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints: Status = Deleted		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Imported Baseline Set
Generic Type:	-	Type:	-	Generic Type:	BaselineSet
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: Status = Available	
Name:	-	Name:	-	Name:	Baseline Set in Archive

Generic Type:	-	Type:	-	Generic Type:	Archive
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Table 7-14 EM205_01_05 Export Import Baseline Set

7.5.6 EM205_01_06 Establish Baseline Sets

Engineering Method: US205_Establish Baseline Sets_EM205_01_06		
Purpose: Baseline all the configuration items included in the selected baseline set definition.		
Comments: -		
Pre-Condition	**Engineering Activity**	**Post-Condition**
Configuration Items are available	05. Triggered by a a request to establish a baseline the SEE shows a list of all Baseline Set Definitions. 06. Select a baseline set definition and create an instance i.e. a Baseline Set and "open" it. 07a. Select a config item from the list of config items and request from CM tool the list of assigned and open CR. 07b. The CM tool provides a list of all open CR for the requested config item. 07c. The SEE checks open CR. If there are no CR open request the associated tool to create a baseline for config item. 07d. The tool baselines the requested config item and generates a new version number. The tool provides the new version number. 07e. The SEE store in the Baseline Set the version number for the config item. 08a. The systems engineer opens a config item in the dedicated tool and selects from an available	Configuration items as defined by the baseline set definition are baselined.

		baseline. The tool requests from the SEE to register this baseline in the open baseline set. 08b. The user selects an available baseline. 08c. The SEE stores the version number with the requested config item in the baseline set. 09. The systems engineer checks for completeness of the baseline set. 10. If all config items are baselined the Systems Engineer closes the baseline. 11. The systems engineer requests a list of all available baseline sets with associated information. 12. The systems engineer requests a list of all config item with associated information for a selected baseline set. 13. The systems engineer selects a dedicated config item and requests to open the baseline of it in the associated tool. 14. The associated tool opens the config item for read only.		
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	-	Name:	-	Name:	NewBaselineSet
Generic Type:	-	Type:	-	Generic Type:	BaselineSet
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: Status = Created	
Name:	ProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-

Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Definition	Name:	-	Name:	-
Generic Type:	BaselineSetDefinition	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-15 EM205_01_06 Establish Baseline Sets

7.5.7 EM205_01_07 List Revision History

Engineering Method: US205_List Revision History_EM205_01_07		
Purpose: To list the revisions and change requests implemented for a dedicated baseline set.		
Comments: -		
Pre-Condition	**Engineering Activity**	**Post-Condition**
Baseline set definitions are available and baseline sets are established.	01. The user requests to generate a revision history. 02. The user selects the baseline set from a list of all baseline sets of a dedicated baseline set definition. 03. The SEE creates the report and requests for each configuration item of the baseline set the relevant change requests. 04. The change management tool extracts all relevant change requests. 05. The SEE collects the change requests and requests further information for each. 06. The change management tool extracts and provides the information for the given change request. 07. The SEE updates the report	N/A

		with the received information.			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Definition	Name:	-	Name:	-
Generic Type:	BaselineSetDefinition	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Base	Name:	-	Name:	-
Generic Type:	BaselineSet	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	RevisionHistory
Generic Type:	-	Type:	-	Generic Type:	Report
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	CurrentChangeRequestDB	Name:	-	Name:	-
Generic Type:	ChangeRequestList	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	ProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-16 EM205_01_07 List Revision History

7.6 PA205_02 - Change Control

7.6.1 EM205_02_01 Manage Change Request

Engineering Method: US205_Manage Change Request_EM205_02_01		
Purpose: To manage the implementation of changes to the data in a controlled manner.		
Comments: -		
Pre-Condition	**Engineering Activity**	**Post-Condition**
Change control is activated for the project. Baseline Sets are available i.e. artefacts are baselined.	01. Create the Change Request and fill in relevant information. 02. The SEE prompts the user to select a baseline set definition. 03. The SEE prompts the user to select a baseline set. 04. The SEE prompts the user to select a configuration item and provides the selected one. 05. The change management completes the change request with a reference to the selected configuration item and baseline. 06. The change management requests the user with configuration control role from the UAM. 07. The UAM extracts the user with configuration management role from the database. 08. The change management requests to send an email to the configuration manager. 09. The emailer prepares the email and requests the email address for the given user from the UAM. 10. The UAM extracts the requested information form the database. 11. The emailer sends the email to the requested user and reports the status. 12. The change management reports the status. 13. The configuration manager assigns a task to a dedicated user.	Changes are incorporated into the database

	The change management tool requests a list of users from the UAM. 14. The UAM extracts a list of users and provides the list. 15. The change management tool prompts the configuration manager to select a user. The tool requests a email to send to the selected user. 16. The emailer prepares the email and requests the email address for the given user from the UAM. 17. The UAM extracts the requested information form the database. 18. The emailer sends the email to the requested user and reports the status. 19. The change management reports the status. 20. The user has completed the task and sets the status of the change request accordingly. The change management requests the user with configuration control role from the UAM. 21. The UAM extracts the user with configuration management role from the database. 22. The change management requests to send an email to the configuration manager. 23. The emailer prepares the email and requests the email address for the given user from the UAM. 24. The UAM extracts the requested information form the database. 25. The emailer sends the email to the requested user and reports the status. 26. The change management reports the status. 27. The configuration manager	

	wants to commission the change to the dedicated user. 28. The change management tool requests a list of selected configuration items from the SEE. 29. The SEE prompts the user to select one of the available baseline set definitions. 30. The SEE provides the list of one or more selected configuration items. 31. For each configuration item the change management tool requests from the UAM a list of users having access to the configuration item. 32. The UAM extracts a list of users having access rights to the given configuration items. The UAM provides a list of valid users. 33. The change management tool prompts the user to select the user, who should be commissioned to change the configuration item. 34. The change management tool requests the UAM to set access rights for the selected user. 35. The UAM sets access rights to the given configuration item for the requested user. 36. 37. The emailer prepares the email and requests the email address for the given user from the UAM. 38. The UAM extracts the requested information form the database. 39. The emailer sends the email to the requested user and reports the status. 40. The change management reports the status.	
Notes: -	Notes: -	Notes: -
Artefacts Required as inputs of the Activities	**Artefacts used internally within the Activities**	**Artefacts Provided as outputs of the Activities**

		(optional)			
Name:	-	Name:	-	Name:	
Generic Type:	-	Type:	-	Generic Type:	
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:		Name:	-	Name:	-
Generic Type:		Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	CurrentProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	UserDB	Name:	-	Name:	-
Generic Type:	SEEUserManagementDB	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	CurrentChangeRequestDB	Name:	-	Name:	-
Generic Type:	ChangeRequest List	Type:	-	Generic Type:	-

Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-17 EM205_02_01 Manage Change Request

7.7 PA205_03 - User and User Access Management

7.7.1 EM205_03_01 Manage Users

Engineering Method: US205_Manage Users_EM205_03_01		
Purpose: The engineering method covers the different activities to manage the users and user groups for the SEE. User groups may be created, deleted and for each CI the access can be defined. Users may be created, deleted and allocated to a defined user group.		
Comments: -		
Pre-Condition	**Engineering Activity**	**Post-Condition**
N/A	01. The SEE Administrator creates a user group within the UAM and requests a list of all SEE tools. 02. The SEE provides a list of tools known to the SEE. 03. The UAM requests from each tool a list of available configuration items. 04. The tool provides a list of all available configuration items. 05. The SEE Admin selects for each config item the access rights for the group. 06. The UAM stores the access rights information in the database. 07. The SEE Administrator creates a new user. 08. The SEE Admin allocates the new user to a pre-defined user group. 09. The SEE Administrator removes a user from the database. 10. The SEE Administrator user group from the data base.	User groups and users are defined, known and permissions are allocated.

		11. The SEE Administrator changes the access rights for a dedicated config item already allocated to a user group.			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	UserDB	Name:	-	Name:	-
Generic Type:	SEEUserManagementDB	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	UserAdmin	Name:	-	Name:	-
Generic Type:	SEEUserReference	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	NewGroup
Generic Type:	-	Type:	-	Generic Type:	SEEUserGroup
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	NewUser
Generic Type:	-	Type:	-	Generic Type:	SEEUserReference
Required Properties:	-	Properties:	-	Provided Properties:	

Description & Interoperability Additional Constraints: -	Description: -	Description & Interoperability Additional Constraints:

Table 7-18 EM205_03_01 Manage Users

7.7.2 EM205_03_02 Load Artefact

Engineering Method: US205_Load Artefact_EM205_03_02		
Purpose: The user loads an artefact of the activated baseline.		
Pre-Condition	**Engineering Activity**	**Post-Condition**
A dedicated baseline is activated from the current trunk or the work.	01. The user requests to show all the available configuration items (or artefacts). 02. The user selects the configuration item he wants to load. 03. The SEE determines the active baseline for the given configuration item. 04. The tool checks the baseline and requests the access mode. 05. The SEE requests the access mode from the UAM tool for the currently logged in user. 06a. The UAM determines the access mode for the requested baseline of the configuration item. 06b. The UAM determines if the item is locked i.e. in use by other users. 07. The SEE checks the access mode and returns in case of access denied or access read only. If access is provided the SEE checks if changes are allowed. 08. The CM checks if for the given configuration item change control is activated. 09. The SEE returns modify access if change control is not yet activated otherwise it requests a list of open change requests. 10. The CM tool provides a list of	Access to the artefact is denied if access is denied. The artefact is loaded in edit mode or in read mode.

		all open CR for the requested config item. 11. The SEE checks open CR. If there are CR open the SEE returns modify access for the given configuration item. Otherwise, it returns read only access. 12. If the receives no access the tool does not load the baseline of the configuration item and informs the user. 13. If the tool receives read only access the tool loads the given baseline of the configuration item in read only mode. 14. The tool receives full access and loads the configuration item in read write mode.			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	CurrentChangeRequestDB	Name:	-	Name:	-
Generic Type:	ChangeRequestList	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	CurrentUser	Name:	-	Name:	-
Generic Type:	SEEUserReference	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Name:	SomeArtefact	Name:	-	Name:	-
Generic Type:	SEE Generic Artefact	Type:	-	Generic Type:	-
Required Properties:	Status Status	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints: Status = Unloaded		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	LoadedArtefact
Generic Type:	-	Type:	-	Generic Type:	SEE Generic Artefact
Required Properties:	-	Properties:	-	Provided Properties:	Status Status
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: Status = Loaded	
Name:	UserDB	Name:	-	Name:	-
Generic Type:	SEEUserManagementDB	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-19 EM205_03_02 Load Artefact

7.7.3 EM205_03_03 Sign On SEE and create New Artefact

Engineering Method: US205_Sign On SEE and create New Artefact_EM205_03_03		
Purpose: The engineering method covers the sign on process to the SEE and the creation of a new artefact in a SEE tool.		
Pre-Condition	**Engineering Activity**	**Post-Condition**
The user is known in the UAM and allocated to a user group. The user is not logged in.	00. The requests to load a project. The SEE requests general access at the UAM. 01. The user signs in with user	The user is logged in to the SEE and a new artefact is created. The access rights for the artefact are assigned in the UAM.

	name and password at the SEE. 02. The UAM checks if the user is registered in its DB and provides the access status. 03. The SEE checks the access mode. 04. In case the user has no access, the SEE informs the user and quits the process. 05a. If access is granted the SEE sets the user as current user on the machine. 05b. If access is granted the SEE sets the user as current user on the machine. 06. The SEE informs the user about successful sign on. 07. The SEE checks the access mode and returns in case of access denied. If access is provided the SEE checks if changes are allowed. 07. The SEEUser requests to create a new artefact in a tool. The tool requests create access. 08. The CM checks if for the given configuration item change control is activated. 08. The SEE processes the create access request and requests create access mode for the current user from UAM. 09. The SEE returns modify access if change control is not yet activated otherwise it requests a list of open change requests. 09. The UAM checks if the given user has a right to create an artefact in the given tool. 10. The CM tool provides a list of all open CR for the requested config item. 11. The SEE checks open CR. If there are CR open the SEE returns access to create an artefact for the	

		parent configuration item. Otherwise, it returns no create access. 12. The tool quits the operation if create access is not provided. 13. The tool creates the artefact and registers the config item at the UAM. 14. The UAM adds the new configuration item to its database. 15. The UAM assigns the access rights to new configuration item (different policies possible) and reports the status. 16. The tool opens the artefact in edit mode.			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	UserDB	Name:	-	Name:	-
Generic Type:	SEEUserManagementDB	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	CurrentUser	Name:	-	Name:	-
Generic Type:	SEEUserReference	Type:	-	Generic Type:	-
Required Properties:	Status Status	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints: Status = Unassigned		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	CurrentUserSet
Generic Type:	-	Type:	-	Generic Type:	SEEUserReferen

					ce
Required Properties:	-	Properties:	-	Provided Properties:	Status Status
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: Status = LoggedIn	
Name:	-	Name:	-	Name:	NewArtefact
Generic Type:	-	Type:	-	Generic Type:	SEE Generic Artefact
Required Properties:	-	Properties:	-	Provided Properties:	Status
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: Status = Created	

Table 7-20 EM205_03_03 Sign On SEE and create New Artefact

7.8 PA205_04 - Process Automation

7.8.1 EM205_04_01 Generate Tool Inventory

Engineering Method: US205_Generate Tool Inventory_EM205_04_01					
Purpose: Generates an inventory list of all the tools registered with the SEE providing detailed version information and configuration.					
Comments: -					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Tools are registered in the SEE		01. The user requests the SEE to generate an inventory list. 02. The SEE generates a report and requests each SEE tool to provide its version information 03. The SEE tool extracts the version information and provides the information to the SEE. 04. The SEE collects the version information from all the SEE tools. 05. The SEE writes the received version information to the report.		N/A	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	-	Name:	-	Name:	ToolInventoryList
Generic Type:	-	Type:	-	Generic Type:	Report
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Table 7-21 EM205_04_01 Generate Tool Inventory

7.8.2 EM205_04_02 Save All

Engineering Method: US205_Save All_EM205_04_02					
Purpose: Save all the changed data throughout the SEE for a consistent view.					
Comments: -					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
A project is loaded into the workspace and changes are performed.		01. The user selects the save all operation at some tool. The tool forwards the request to the SEE. 02. The SEE extracts the list of registered tools and requests each tool to save all its data. 03. The tool save all its data and provides an operation status. 04. The SEE collects the status of each tools and generates a combined one. 05. The triggering tool displays the combined status. 28. The change management tool requests a list of selected configuration items from the SEE.		All modifications across all the tools are saved to the database.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Changed Data	Name:	-	Name:	-
Generic Type:	ConfigurationItem	Type:	-	Generic Type:	-
Required Properties:	IsModified IsModified	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints: IsModified = Data Modified		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Saved Data
Generic Type:	-	Type:	-	Generic Type:	ConfigurationItem
Required	-	Properties:	-	Provided	IsModified

Properties:				Properties:	IsModified
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: IsModified = Data saved	
Name:	ProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-22 EM205_04_02 Save All

7.8.3 EM205_04_03 Search Data

Engineering Method: US205_Search Data_EM205_04_03		
Purpose: To search the data across the SEE tools.		
Comments: -		
Pre-Condition	**Engineering Activity**	**Post-Condition**
N/A	01. The user requests to search for elements including a given search string. 02. The user defines the search string using regular expressions. 03. The SEE tool searches the database for occurrences of the given search string and provides a list of all findings. 04. The SEE collect the findings. 05. The SEE displays all the findings. 06. The user is interested in which tool, configuration item and baseline a dedicated element is included. 07. The user defines the ID optionally using wildcards. 08. The SEE tool searches the database and all baselines for occurrences of an element with the	N/A

		given ID. 09. The SEE collect the findings. 10. The SEE displays all the findings.			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	ProjectFile	Name:	-	Name:	-
Generic Type:	SEE Project File	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Artefact	Name:	-	Name:	-
Generic Type:	SEE Generic Artefact	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-23 EM205_04_03 Search Data

7.9 PA206_01 - Maintain Traceability

7.9.1 EM206_01_01 Retrieve Valid Traces

Engineering Method: US206_Retrieve Valid Trace_EM206_01_01					
Purpose: To get the valid links of the system engineering environment for accessibility.					
Comments: -					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Links of valid artefacts are available in the system engineering environment.		01. Requests for the outgoing links of all artefacts, of which the status is valid 02. Illustrate the traceability network within SEE by displaying the artefacts and the corresponding links		Valid Links are accessible in the system engineering environment.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	available Links	Name:	-	Name:	-
Generic Type:	Link	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	valid Links
Generic Type:	-	Type:	-	Generic Type:	LinkList
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	

Table 7-24 EM206_01_01 Retrieve Valid Traces

7.9.2 EM206_01_02 Analyse Trace

Engineering Method: US206_Analyse Trace_EM206_01_02					
Purpose: To analyse the environment for suspect links and mark them respectively.					
Comments: -					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
Valid Links are accessible in the system engineering environment.		01. Analyse if artefacts within SEE were changed 02. Check for links, which are affected by a changed target 03. Mark the affected links as suspect 04. Within the SEE the suspected links are highlighted to be identifiable by the user		Links with changed target artefact were marked as suspect.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	valid Links	Name:	-	Name:	-
Generic Type:	LinkList	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	updated Links
Generic Type:	-	Type:	-	Generic Type:	Link
Required Properties:	-	Properties:	-	Provided Properties:	LinkSource LinkTarget ArtefactStatus
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: ArtefactStatus = "valid"	

Table 7-25 EM206_01_02 Analyse Trace

7.10 PA206_02 - Monitor Development Progress

7.10.1 EM206_02_01 Perform Coverage Analysis

Engineering Method: US206_Perform Coverage Analysis_EM206_02_01					
Purpose:					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
		01. get the list of configuration item for the selected baseline set. 02. Analyse the provided requirements for in-going links of the type refine and decompose. 03. Analyse the requirements of the provided requirements for in-going links of the type satisfy. 04. Analyse the requested requirements for the out-going links of the type refine and decompose. 05. Analyse the provided requirements for in-going links of the type verify. 06. Analyse the design artefacts for out-going links of the type satisfy. 07. Analyse the design artefacts for in-going links of the type derive. 08. Analyse the change actions for in-going links of the type modify. 09. Collect and prepare the results of the performed coverage analyses. 10. provide the coverage results.		
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	-	Name:	-	Name:	Coverage Report
Generic Type:	-	Type:	-	Generic Type:	Report
Required Properties:	-	Properties:	-	Provided Properties:	ReportType

Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: ReportType = "Coverage Report"	
Name:	valid Links	Name:	-	Name:	-
Generic Type:	Link	Type:	-	Generic Type:	-
Required Properties:	ArtefactStatus	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-26 EM206_02_01 Perform Coverage Analysis

7.11 PA206_03 - Perform Verification

7.11.1 EM206_03_01 Create Verification Objective

Engineering Method: US206_Create Verification Objective_EM206_03_01		
Purpose: The verification objective and traceability to the selected requirement will be established.		
Comments: The creation of the empty verification objective and the traceability shall be generated automatically.		
Pre-Condition	**Engineering Activity**	**Post-Condition**
The set of requirements to be verified is available.	01. Request the List of Requirements selected by the VVEngineer 02. display previous view on the requirements selected from the VVEngineer 03. Requirement for the definition of the verification objective selected by the VVEngineer 04a. Request for the detailed information of the selected requirement 04b. Request for design artefacts allocated to selected Requirement 04c. Create the empty verification objective for selected requirement 05. Generate Verification Objective based on requirement and design information 06. Establish "derive"-Link between	The verification objective of a selected requirement was created and a link to the selected requirement (link type = derived) was established.

		Verification Objective and Requirement			
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Requirements	Name:	-	Name:	-
Generic Type:	Requirement	Type:	-	Generic Type:	-
Required Properties:	ID ObjectText RequirementType	Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	Design Artefacts	Name:	-	Name:	-
Generic Type:	SystemModel	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Verification Objectives
Generic Type:	-	Type:	-	Generic Type:	VerificationObjective
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	Link From Objective To Requirement
Generic Type:	-	Type:	-	Generic Type:	derive

Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: LinkTarget = "Requirement" ArtefactStatus = "valid" LinkSource = "Verification Objective"	

Table 7-27 EM206_03_01 Create Verification Objective

7.11.2 EM206_03_02 Create Verification Case

Engineering Method: US206_Create Verification Case_EM206_03_02					
Purpose: The verification case and traceability to verification objective and requirement will be created.					
Comments: The creation of the empty verification case and the traceability shall be generated automatically.					
Pre-Condition		**Engineering Activity**		**Post-Condition**	
The verification objectives are created and linked to the requirements (link type = derived).		01. Create an empty verification case for the selected objective 02. Establish "derive"-Link between verification cases and objective 03. Generate the verification cases authorised by the VVEngineer 04. Get the requirements allocated to the selected verification objectives 05. Establish automatically the "verify"-Links between verification cases and requirements		The verification case for a selected verification objective was created and links to the verification objectives (link type = derived) and to the allocated requirements (link type = verify) were established.	
Notes: -		Notes: -		Notes: -	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	Verification Objectives	Name:	-	Name:	-
Generic Type:	VerificationObjective	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-

Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	Verification Case
Generic Type:	-	Type:	-	Generic Type:	VerificationCase
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints:	
Name:	-	Name:	-	Name:	Link From Case To Requirement
Generic Type:	-	Type:	-	Generic Type:	verify
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: LinkSource = "Verification Case" LinkTarget = "Requirement" ArtefactStatus = "valid"	
Name:	-	Name:	-	Name:	Links From Case To Objectives
Generic Type:	-	Type:	-	Generic Type:	derive
Required Properties:	-	Properties:	-	Provided Properties:	
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: LinkSource = "Verification Objective" LinkTarget = "Verification Case" ArtefactStatus = "valid"	
Name:	Links From Objectives To Requirements	Name:	-	Name:	-
Generic Type:	derive	Type:	-	Generic Type:	-
Required		Properties:	-	Provided	-

Properties:				Properties:	
Description & Interoperability Additional Constraints: LinkTarget = "Requirement" ArtefactStatus = "valid" LinkSource = "Verification Objective"		Description: -		Description & Interoperability Additional Constraints: -	

Table 7-28 EM206_03_02 Create Verification Case

7.11.3 EM206_03_03 Create Verification Procedure

Engineering Method: US206_Create Verification Procedure_EM206_03_03		
Purpose: The verification procedure steps and traceability to verification case and requirement will be created. The verify links of the procedure will replace the links of the verification case. The ones from the verification case will be deleted.		
Comments: The creation of the empty verification procedure steps and the traceability shall be generated automatically.		
Pre-Condition	**Engineering Activity**	**Post-Condition**
The verification cases are created and are linked to the verification objectives (link type = derive) and to the allocated requirements (link type = verify). The previous verify links from the selected verification case to the requirements were deleted.	01. Create an empty object for verification procedure steps 02. Create an "decompose"-Link between verification case and the verification procedure 03. Generate a verification procedure based on verification cases 04. Get the requirements within the VVMgmt allocated to the selected verification cases 05. Create automatically "verify"-Link between verification cases and requirements 06. Get the verification cases within the VVMgmt allocated to the selected to the verification procedure steps. 07. Remove the "verify"-Links between verification cases and requirements, because they are replaced by the verification procedure	The verification procedure steps for a selected verification case was created and links to the verification cases (link type = decompose) and to the allocated requirements (link type = verify) were established.
Notes: -	Notes: -	Notes: -

Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities (optional)**		**Artefacts Provided as outputs of the Activities**	
Name:	-	Name:	-	Name:	VerificationProcedure
Generic Type:	-	Type:	-	Generic Type:	Document
Required Properties:	-	Properties:	-	Provided Properties:	DocumentType
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: DocumentType = "VerificationProcedure"	
Name:	Verification Case	Name:	-	Name:	-
Generic Type:	VerificationCase	Type:	-	Generic Type:	-
Required Properties:		Properties:	-	Provided Properties:	-
Description & Interoperability Additional Constraints:		Description: -		Description & Interoperability Additional Constraints: -	
Name:	-	Name:	-	Name:	ProcedureLink
Generic Type:	-	Type:	-	Generic Type:	Link
Required Properties:	-	Properties:	-	Provided Properties:	LinkSource LinkTarget ArtefactStatus
Description & Interoperability Additional Constraints: -		Description: -		Description & Interoperability Additional Constraints: LinkSource = "Verification Procedure" LinkTarget = "Verification Case" ArtefactStatus = "valid"	

Table 7-29 EM206_03_03 Create Verification Procedure

7.12 PA206_04 - Analyse Impact of Change

7.12.1 EM206_04_01 Analyse Impact of Change

<table>
<tr><th colspan="6">Engineering Method: US206_Analyse Impact of Change_EM206_04_01</th></tr>
<tr><td colspan="6">Purpose: To get an overview for the impact of a change. All affected artefacts are reported.</td></tr>
<tr><td colspan="6">Comments: The analysis is performed over all levels of development.</td></tr>
<tr><th colspan="2">Pre-Condition</th><th colspan="2">Engineering Activity</th><th colspan="2">Post-Condition</th></tr>
<tr><td colspan="2">Change proposal available and the traceability established during development.</td><td colspan="2">01. Provide artefact IDs, of source artefacts linked to the requirement to be changed.
02. Illustrate the affected artefacts by an impact tree.
03. Provide the context information in which a selected artefacts interacts.
04. Display the context information requested for the specific artefact.</td><td colspan="2">The change impact is illustrated in a report.</td></tr>
<tr><td colspan="2">Notes: -</td><td colspan="2">Notes: -</td><td colspan="2">Notes: -</td></tr>
<tr><th colspan="2">Artefacts Required as inputs of the Activities</th><th colspan="2">Artefacts used internally within the Activities
(optional)</th><th colspan="2">Artefacts Provided as outputs of the Activities</th></tr>
<tr><td>Name:</td><td>validLinks</td><td>Name:</td><td>-</td><td>Name:</td><td>-</td></tr>
<tr><td>Generic Type:</td><td>Link</td><td>Type:</td><td>-</td><td>Generic Type:</td><td>-</td></tr>
<tr><td>Required Properties:</td><td>LinkSource
LinkTarget
ArtefactStatus</td><td>Properties:</td><td>-</td><td>Provided Properties:</td><td>-</td></tr>
<tr><td colspan="2">Description & Interoperability Additional Constraints:
ArtefactStatus = "valid"
LinkTarget = "changed target artefact or child of changed target artefact"</td><td colspan="2">Description:
-</td><td colspan="2">Description & Interoperability Additional Constraints:
-</td></tr>
<tr><td>Name:</td><td>-</td><td>Name:</td><td>-</td><td>Name:</td><td>Change Impact</td></tr>
<tr><td>Generic Type:</td><td>-</td><td>Type:</td><td>-</td><td>Generic Type:</td><td>Report</td></tr>
<tr><td>Required Properties:</td><td>-</td><td>Properties:</td><td>-</td><td>Provided Properties:</td><td>ReportType</td></tr>
</table>

Description & Interoperability Additional Constraints: -	Description: -	Description & Interoperability Additional Constraints: ReportType = "Change Impact"

Table 7-30 EM206_04_01 Analyse Impact of Change

8 Annex III: Requirements

In the following the requirements that need to be fulfilled by the SEE are defined. Brackets are used in the requirements text in order to indicate terms that are further described in the glossary.

8.1 General Requirements

8.1.1 General Process Automation Requirements

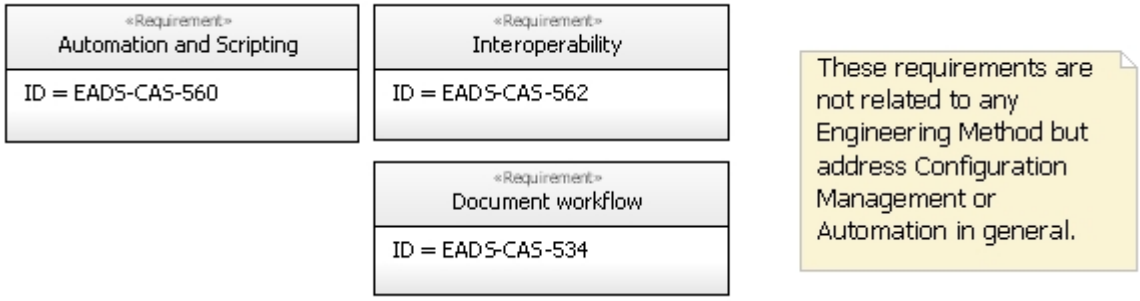

Figure 8-1 General Process Automation Requirements

EADS-CAS-560

The SEE shall provide the APIs to enable cross-tool automation. The APIs for automation and scripting shall allow

- to access and to modify all the model elements and their attributes,
- to manage the model elements and their associations (e.g. add, delete),
- to access and to modify SEE and tool options and
- to access features of the tools itself like menus and toolbars.

Add. Info: The SEE provides the user the capability to create scripts and applications for customization and automation of the tools. Therefore, the SEE enables the user to adapt the tool environment to its own needs, to tailor the SEE according project needs and to extend the SEE to support own processes and methods.

EADS-CAS-534

The SEE shall store with each artefact the status of that artefact, which is for generic artefacts and documents:

- Created
- In Work
- Checking
- In Review
- Approved
- Obsolete
- Deleted

For change requests and problem reports the SEE shall enable the user to setup an own status according to a project specific configuration management plan.

EADS-CAS-562

The SEE shall provide interoperability capabilities between the different integrated tools. Tools implementing and exposing similar functions shall be exchangeable and shall provide identical API for automation.

Add. Info: Efficient SE is only possible, when the tools are integrated. This avoids unnecessary data-exchange problems (formats, handling, etc.).

8.1.2 General Traceability Requirements

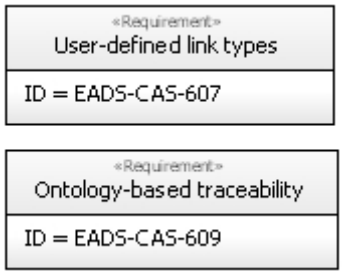

Figure 8-2 General Traceability Requirements

EADS-CAS-607

The SEE shall provide the user the possibility to define additional [link types].

EADS-CAS-609

The SEE shall provide the capability to discover possible [links] to other artefacts based on the defined [ontologies].

Add. Info: For example while creating new artefacts or running an analysis.

8.2 PA202_01 - Perform Functional Safety Analysis

8.2.1 EM202_01_01 - Transfer Model Data for Fault Tree Analysis

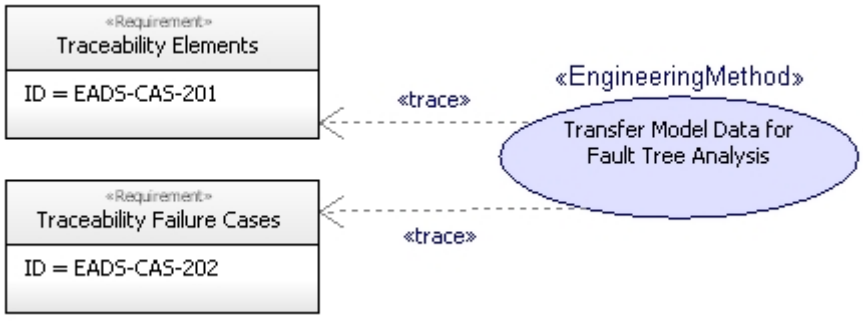

Figure 8-3 EM202_01_01 Transfer Model Data for Fault Tree Analysis

EADS-CAS-201

The SEE shall allow to trace from basic elements in a Fault Tree Analysis tool to use cases, blocks and primitive operations of a SysML model.

EADS-CAS-202

The SEE shall allow to trace from failure cases in a Fault Tree Analysis tool to tagged values of use cases, blocks and primitive operations of a SysML model.

8.2.2 EM202_01_02 - Check System Model and Fault Tree Consistency

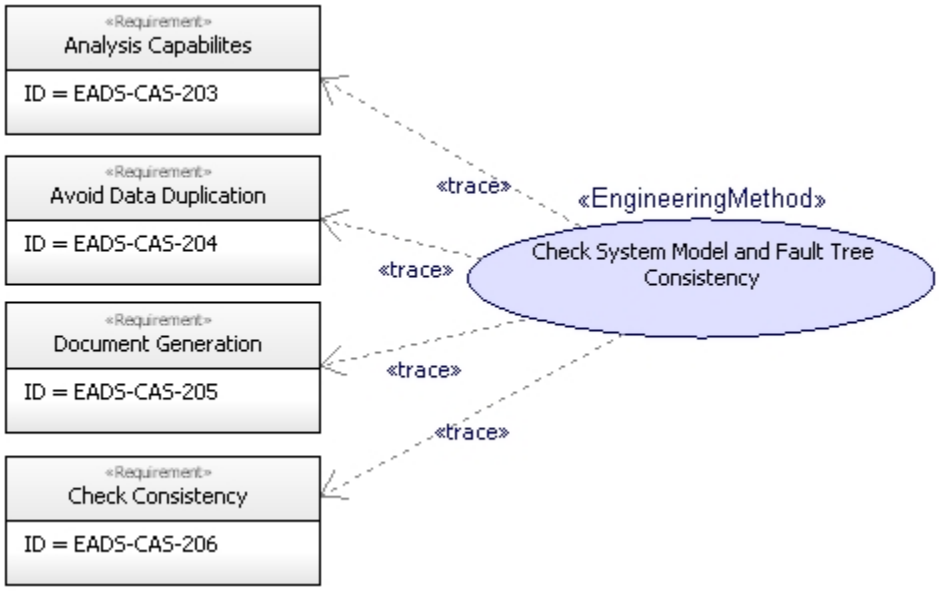

Figure 8-4 EM202_01_02 Check System Model and Fault Tree Consistency

EADS-CAS-203

The SEE shall provide safety analysis capabilities related to system design. The system design model shall be the input and baseline for the analysis.

EADS-CAS-204

The SEE shall ensure consistency of the data between System and Fault Tree Analysis model. Duplication of data shall be avoided.

EADS-CAS-206

The SEE shall allow to compare baselines of system (SysML) model and faul tree models. The SEE shall highlight inconsistencies using different colours. Per user setting, matching parts shall be suppressed.

EADS-CAS-205

The SEE shall provide cross system (SysML) model and fault tree model report and document generation capabilities on the basis of a selected baselines.

8.3 PA203_01 - Product Family Domain Engineering

8.3.1 EM203_01_01 - Develop Domain System Requirements

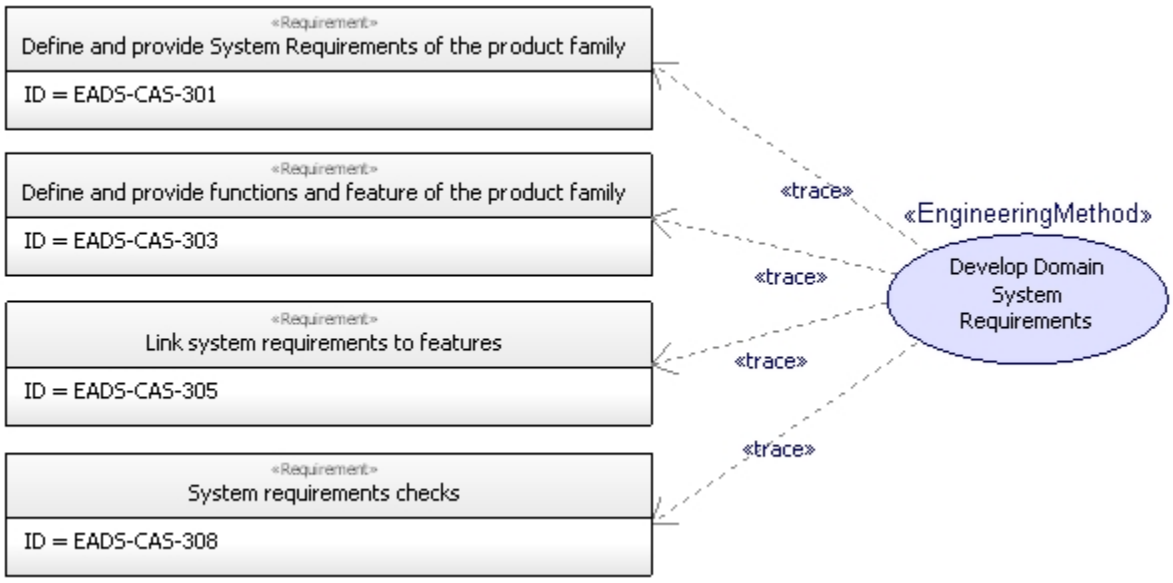

Figure 8-5 EM203_01_01 Develop Domain System Requirements

EADS-CAS-303

The SEE shall provide means to define common and variable functions and features. Relations and constraints as for e.g. requires or excludes between functions, features and feature options shall be defined.

EADS-CAS-305

The SEE shall provide the capability to link system requirements to common features or feature options. Common system requirements shall be linked to common features. Variable system requirements shall be linked to features options.
The linking information between requirements and features/options shall be accessible by other SEE tools.

EADS-CAS-308

The SEE shall check feature model and system requirements for completeness and consistency:

- if there are requirements without an allocated feature or option.
- if requirements are allocated to deleted features or options (orphan).
- if requirements are allocated to changed features or options (suspect).

EADS-CAS-301

The SEE shall define common and variable system requirements. For variable system requirements the SEE shall provide the capability to specify [variation point]s and provide the variability information to other tools.

8.3.2 EM203_01_02 - Develop Reference Architecture

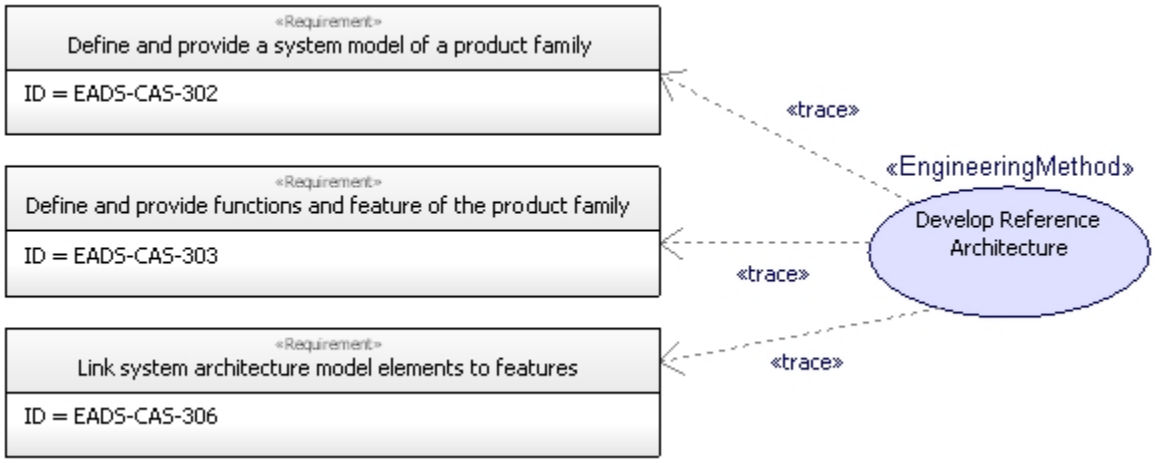

Figure 8-6 EM203_01_02 Develop Reference Architecture

EADS-CAS-306

The SEE shall allocate system architecture model elements to common features or feature options.
Common system model elements shall be linked to common features. Variable system model elements shall be linked to features options.
The allocation information between the elements and features shall be accessible by the SEE tools.

EADS-CAS-303

The SEE shall provide means to define common and variable functions and features. Relations and constraints as for e.g. requires or excludes between functions, features and feature options shall be defined.

EADS-CAS-302

The SEE shall provide the capability to specify the commonality and variability of system model elements, their relations and constraints and [variation points] in the system model.

Add. Info: The SEE shall provide the capability to specify [variation point]s for system models and provide the information to other tools.
Beside variation points the relations and constraints between variable system model elements shall be defined. For example object existence of components and interfaces or thresholds need to be defined.

A convention shall be used to describe and provide variability information of system architecture models. E.g. CVL.

Variation Points shall provide this information:
- <variation type> defining if the [variation point] relates to a function, feature, system requirement, system model.
- a <variation point ID> and <variation name> to name and identify variation points,
- the <variation description> to specify what shall vary,
- the <options-ID> and the possible <option> to define the variation range,
- the <rationale> to describe why a variation point is required,
- the <binding time> definition to describe when the variation shall be resolved: <compile/link time> or <integration time> or <installation time> or <operation time>,
- the <stakeholder> to document the requesting stakeholder of a feature and feature option.
- the <visibility> to define the visibility of the variation, external e.g. for the customer or internal only.

8.4 PA203_02 - Product Realization

8.4.1 EM203_02_01 - Create Product System Requirements

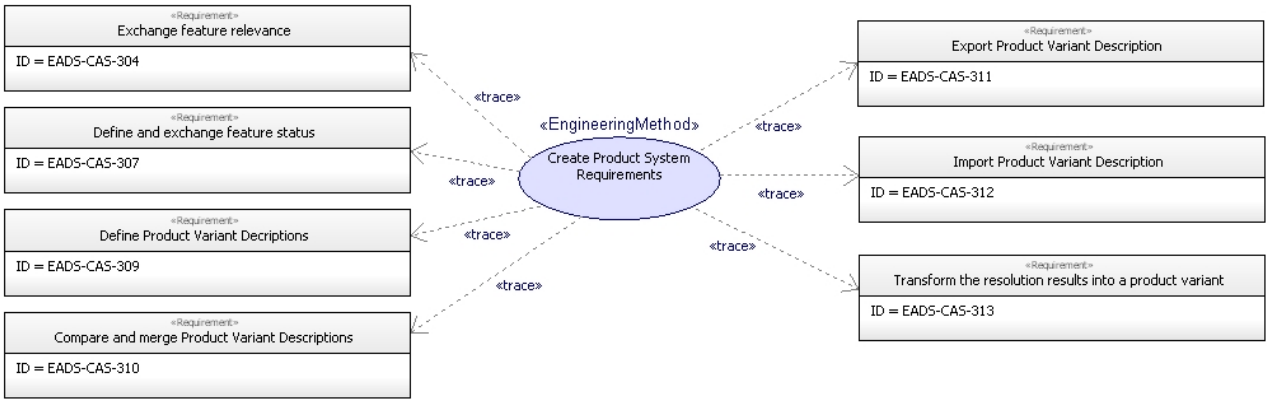

Figure 8-7 EM203_02_01 Create Product System Requirements

EADS-CAS-313

The SEE shall use a [product variant description] to extract the product system requirements and product system model elements from the product family system requirements and system model.
A product requirements database and a product system model shall contain the exported data.
For each creation of a product variant a transformation protocol shall document the selection of system requirements and system model elements. The decisions and [variation point]s why elements or requirements are included or excluded shall be documented.

EADS-CAS-309

The SEE shall guide through the variability resolution of functions, features and feature options, system requirements and system architecture model [variation points]. This shall be done top down (functions, features, requirements, system architecture) considering the feature relevance. Only features with the feature type "available" shall be proposed for resolution.

EADS-CAS-304

The SEE shall exchange the feature relevance information to support variability resolution. For each feature and feature option the relevance assessment information shall be provided:

- feature relevance (High, Medium, Low)
- the feature importance for customer satisfaction (High, Medium, Low)
- the feature technology readiness (High, Medium, Low)
- the costs to provide the feature (High, Medium, Low).

EADS-CAS-310

The SEE shall provide the capability to compare, to indicate differences and to merge product variant descriptions.

EADS-CAS-311

The SEE shall provide the capability to export [product variant description]s in XML format to be human readable to verify the resolution. Baselining information (version, baseline, date) shall be included in the exported file.

EADS-CAS-312

The [product variant description] files shall be importable to create/modify a product variant.

EADS-CAS-307

The SEE shall provide feature status (new, planned, development, future, available).

8.4.2 EM203_02_02 - Create Product System Architecture

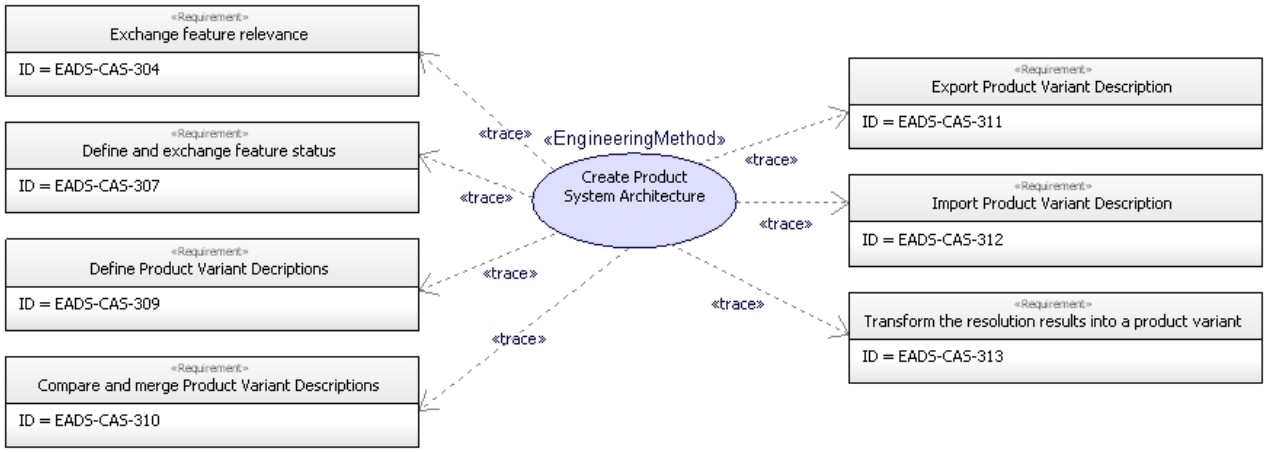

Figure 8-8 EM203_02_02 Create Product System Architecture

EADS-CAS-313

The SEE shall use a [product variant description] to extract the product system requirements and product system model elements from the product family system requirements and system model.
A product requirements database and a product system model shall contain the exported data.
For each creation of a product variant a transformation protocol shall document the selection of system requirements and system model elements. The decisions and [variation point]s why elements or requirements are included or excluded shall be documented.

EADS-CAS-309

The SEE shall guide through the variability resolution of functions, features and feature options, system requirements and system architecture model [variation points]. This shall be done top down (functions, features, requirements, system architecture) considering the feature relevance. Only features with the feature type "available" shall be proposed for resolution.

EADS-CAS-310

The SEE shall provide the capability to compare, to indicate differences and to merge product variant descriptions.

EADS-CAS-311

The SEE shall provide the capability to export [product variant description]s in XML format to be human readable to verify the resolution. Baselining information (version, baseline, date) shall be included in the exported file.

EADS-CAS-312

The [product variant description] files shall be importable to create/modify a product variant.

EADS-CAS-304

The SEE shall exchange the feature relevance information to support variability resolution. For each feature and feature option the relevance assessment information shall be provided:

- feature relevance (High, Medium, Low)
- the feature importance for customer satisfaction (High, Medium, Low)
- the feature technology readiness (High, Medium, Low)
- the costs to provide the feature (High, Medium, Low).

EADS-CAS-307

The SEE shall provide feature status (new, planned, development, future, available).

8.5 PA204_01 - Perform Requirements Engineering

8.5.1 EM204_01_01 - Define Requirements

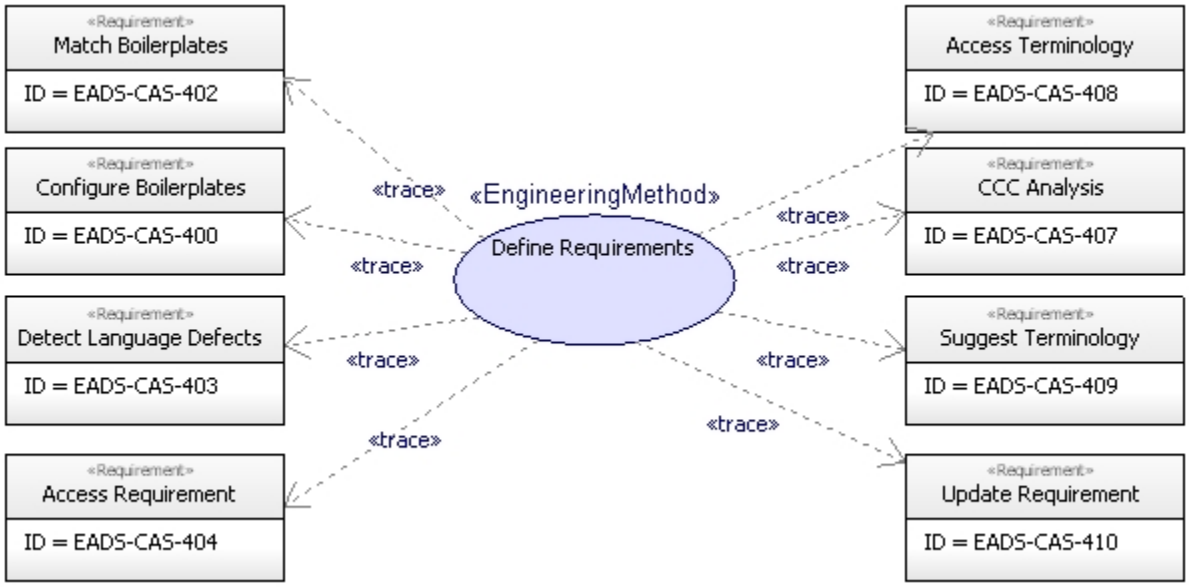

Figure 8-9 EM204_01_01 Define Requirements

EADS-CAS-404

The SEE shall provide support to access the requirement text, history, pre-defined attribute values, and traceability information for a given requirement.

EADS-CAS-410

The SEE shall provide support to update the requirement text of a given requirement in the ReqMgmt tool.

EADS-CAS-407

The SEE shall provide the capability to analyze the consistency and completeness of given requirements sets.

EADS-CAS-408

The SEE shall provide support to access the [controlled vocabulary] of a given domain.

EADS-CAS-402

The SEE shall provide the capability to identify matching boilerplates of a given boilerplate group for a given natural language requirement.

EADS-CAS-409

The SEE shall provide support to suggest terminology during requirement authoring depending on the structure and semantics of the applied boilerplate.

EADS-CAS-400

The SEE shall provide the capability to configure the boilerplate group applicable for requirements quality analysis.

EADS-CAS-403

The SEE shall provide the capability to detect [language defects] in natural-language requirements.

8.5.2 EM204_01_02 - Search Requirements

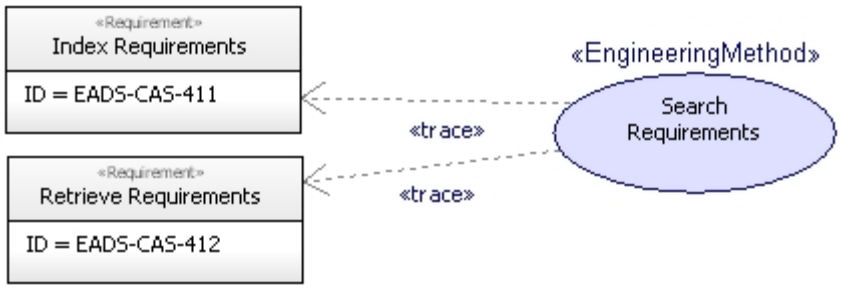

Figure 8-10 EM204_01_02 Search Requirements

EADS-CAS-411

The SEE shall provide the capability to formalize natural language requirement into knowledge structures that allow semantic search.

EADS-CAS-412

The SEE shall provide the capability to search requirements based on the following search criteria:

- requirement types
- used concepts
- similarity with existing requirements

8.5.3 EM204_01_03 - Analyze Requirements Quality

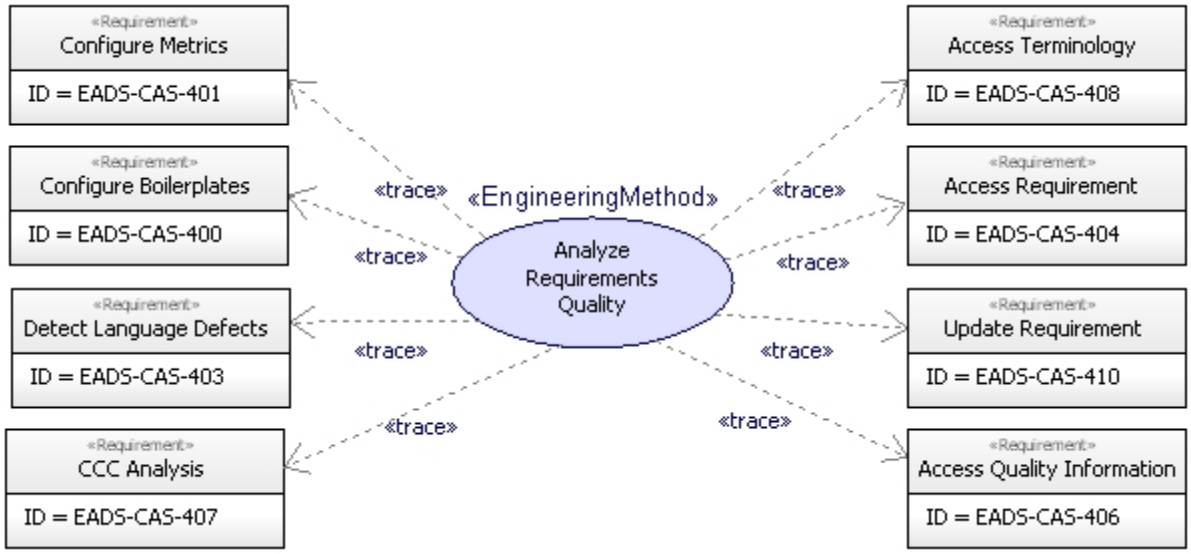

Figure 8-11 EM204_01_03 Analyze Requirements Quality

EADS-CAS-404

The SEE shall provide support to access the requirement text, history, pre-defined attribute values, and traceability information for a given requirement.

EADS-CAS-410

The SEE shall provide support to update the requirement text of a given requirement in the ReqMgmt tool.

EADS-CAS-406

The SEE shall provide support to access the quality information for a given requirements set from the ReqQualityAnalysis tool.

EADS-CAS-407

The SEE shall provide the capability to analyze the consistency and completeness of given requirements sets.

EADS-CAS-408

The SEE shall provide support to access the [controlled vocabulary] of a given domain.

EADS-CAS-401

The SEE shall provide the capability to configure the quality metrics applicable for requirements quality analysis.

EADS-CAS-400

The SEE shall provide the capability to configure the boilerplate group applicable for requirements quality analysis.

EADS-CAS-403

The SEE shall provide the capability to detect [language defects] in natural-language requirements.

8.6 PA205_01 - Configuration Management

8.6.1 EM205_01_01 - Create Baseline Set Definition

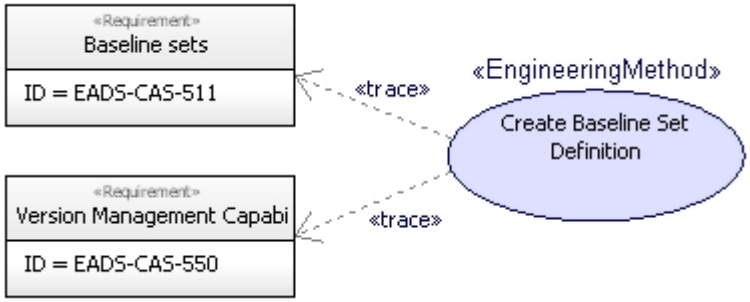

Figure 8-12 EM205_01_01 Create Baseline Set Definition

EADS-CAS-550

The SEE environment shall provide version management capability for tools not supporting those features. The SEE shall provide an API to allow those tools to check out and check in data.

Add. Info: The SEE provides the capabilities and API allowing tools to check out and check in files and data. This allows baselining and change control also for tools not originally having implemented such feature. Furthermore, it keeps the tools simple allowing them to concentrate on their core functionality.

EADS-CAS-511

The SEE shall allow the definition of [Baseline Sets] reflecting the cross-tool scope of a configuration.

Add. Info: When a baseline is established, artefacts shall be in a specific configuration state.

8.6.2 EM205_01_02 - Branch Baseline

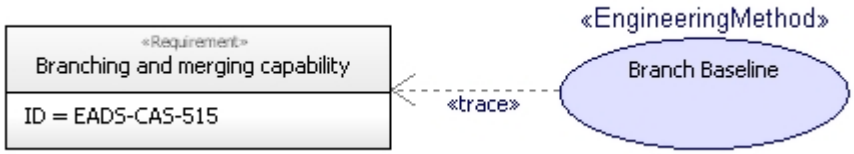

Figure 8-13 EM205_01_02 Branch Baseline

EADS-CAS-515

The SEE shall provide branching and merging capabilities.

Add. Info: Change orders may be allocated to and implemented within a dedicated branch, i.e. the problem is not solved in a parallel branch. That has to be clearly detectable.

8.6.3 EM205_01_03 - List Change History

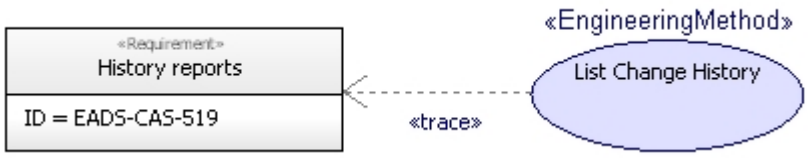

Figure 8-14 EM205_01_03 List Change History

EADS-CAS-519

The SEE shall provide report generation for change management listing for each [Configuration Item] its change history, implemented change orders, the artefacts changed due to a change order and any further open problems linked to the [Configuration Item].

Add. Info: Before/after case, change history (implemented change requests + changed artefacts).

8.6.4 EM205_01_04 - Merge Baselines

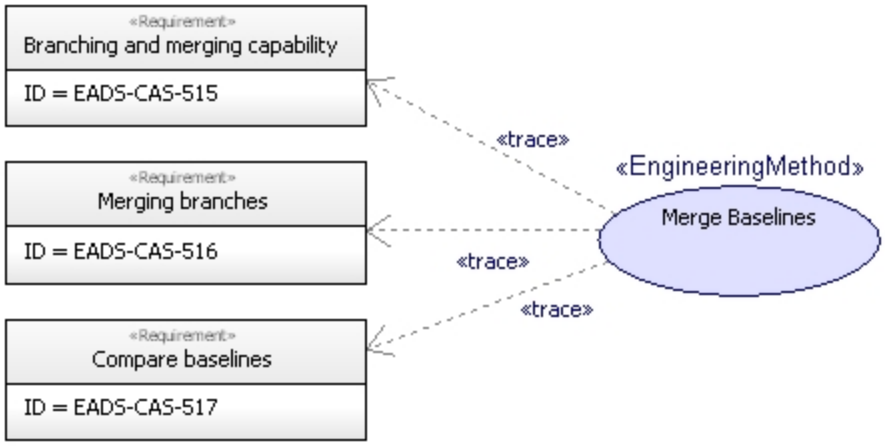

Figure 8-15 EM205_01_04 Merge Baselines

EADS-CAS-517

The SEE shall allow to compare [Baselines] and [Baseline Sets]. The SEE shall highlight differences using different colours. Per user setting, identical parts shall be suppressed.

Add. Info: Identification of changes in the configuration of single CI between two baselines (evolution).

EADS-CAS-516

The SEE shall allow the merging of a branch with the current work stream. The SEE shall automatically solve conflicts i.e. merge the changes of both versions.

EADS-CAS-515

The SEE shall provide branching and merging capabilities.

Add. Info: Change orders may be allocated to and implemented within a dedicated branch, i.e. the problem is not solved in a parallel branch. That has to be clearly detectable.

8.6.5 EM205_01_05 - Export Import Baseline Set

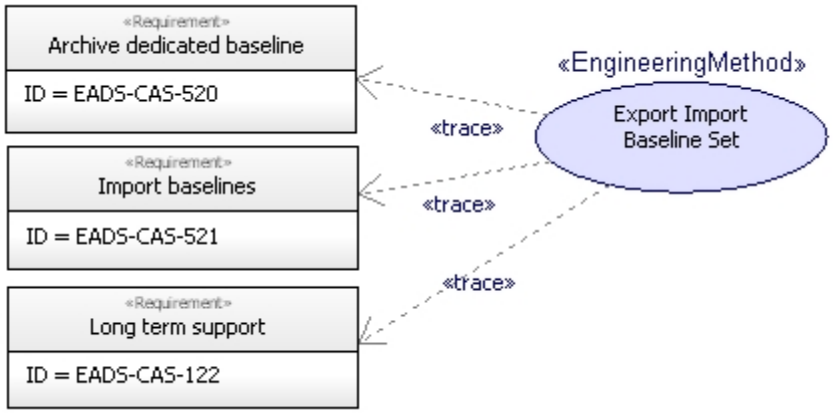

Figure 8-16 EM205_01_05 Export Import Baseline Set

EADS-CAS-520

The SEE shall provide the capability to export dedicated [Baseline Sets] to the local file system e.g. for purpose of releasing a [Baseline Set] to the PLM system or to archive a baseline. The SEE shall assure the long term accessibility of the data e.g. by using standardized formats instead of proprietary data formats.

EADS-CAS-521

The SEE shall provide the capability to import a [Baseline Set] from local file system. The SEE shall be able to import a previously archived [Baseline Set] i.e. to roll back a dedicated product version from the PLM system into the SEE.

Add. Info: This allows developing new variants based on a previous product.

EADS-CAS-122

The SEE shall be robust against evolution of single tools and standardized tool interface definitions (API). The SEE shall provide an upgrade path for databases and tools to provide support for at least 30 years once a project is archived. The SEE shall consider, that projects will not follow each update instead will skip minor or even several major updates. Proprietary file formats shall be avoided.

Add. Info: Standardization of data supports the long term accessibility of data. At least 30 years need to be supported. In case of incompatibilities converters may be considered.

8.6.6 EM205_01_06 - Establish Baseline Sets

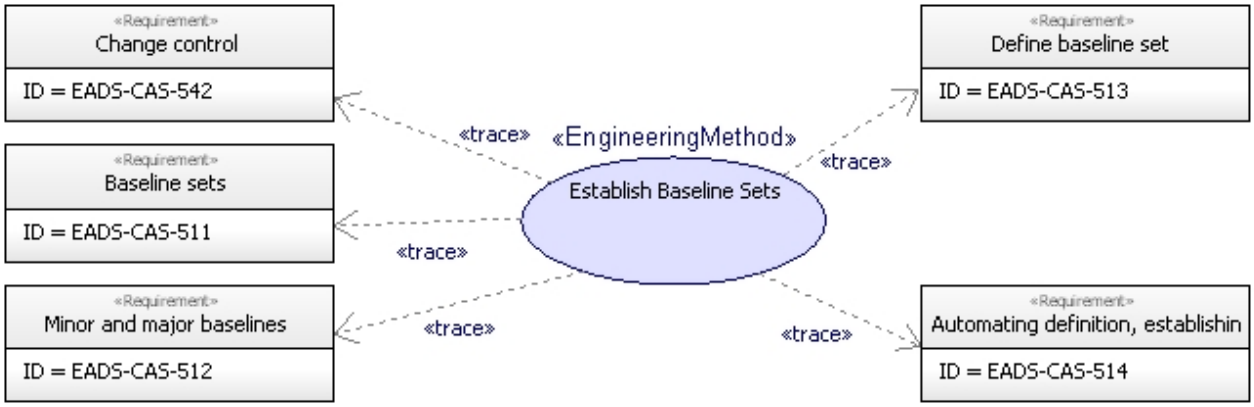

Figure 8-17 EM205_01_06 Establish Baseline Sets

EADS-CAS-511

The SEE shall allow the definition of [Baseline Sets] reflecting the cross-tool scope of a configuration.

Add. Info: When a baseline is established, artefacts shall be in a specific configuration state.

EADS-CAS-512

The SEE shall allow the definition of minor [Baselines] and major [Baselines].

Add. Info: Major baselines are released to PLM, whereas minor baselines are released to the engineering team.

EADS-CAS-514

The SEE shall provide automation (wizards) for establishing a [Baseline Set].

Add. Info: The development activities range from the setup of requirements repositories, the creation of package structures to more complex transformations of existing model data into other views.

EADS-CAS-542

The SEE shall control all identified [Configuration Items] with respect to history, tracing, configuration and change control.

Add. Info: Artefacts include also documents, such as .doc(x), .ppt(x), .xls(x), .jpg, .vsd, ...

EADS-CAS-513

The SEE shall provide screens and dialogues to create a [Baseline Set Definition].

Add. Info: A [Baseline Set Definition] for a system or product variant references all the artefacts that are necessary to build these elements.
The user may also create [Baseline Set Definitions] for subsets of the system elements or with a user-defined scope. In this case, the SEE shall support the user to identify the complete set of artefacts to be included in the [Baseline Set Definition]. For this, the SEE should evaluate the different cross-tool dependencies.

8.6.7 EM205_01_07 - List Revision History

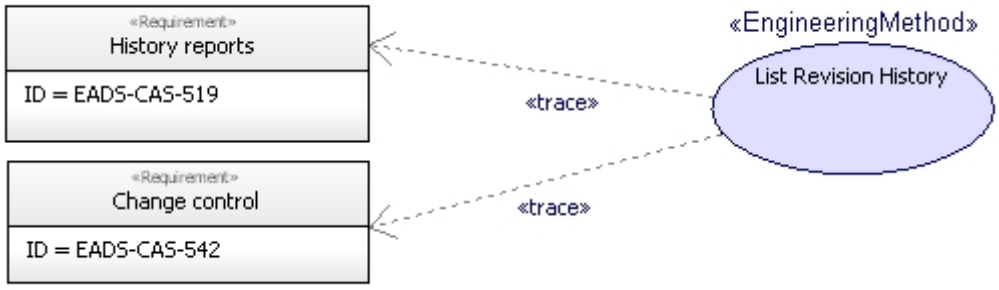

Figure 8-18 EM205_01_07 List Revision History

EADS-CAS-519

The SEE shall provide report generation for change management listing for each [Configuration Item] its change history, implemented change orders, the artefacts changed due to a change order and any further open problems linked to the [Configuration Item].

Add. Info: Before/after case, change history (implemented change requests + changed artefacts).

EADS-CAS-542

The SEE shall control all identified [Configuration Items] with respect to history, tracing, configuration and change control.

Add. Info: Artefacts include also documents, such as .doc(x), .ppt(x), .xls(x), .jpg, .vsd, ...

8.7 PA205_02 - Change Control

8.7.1 EM205_02_01 - Manage Change Requests

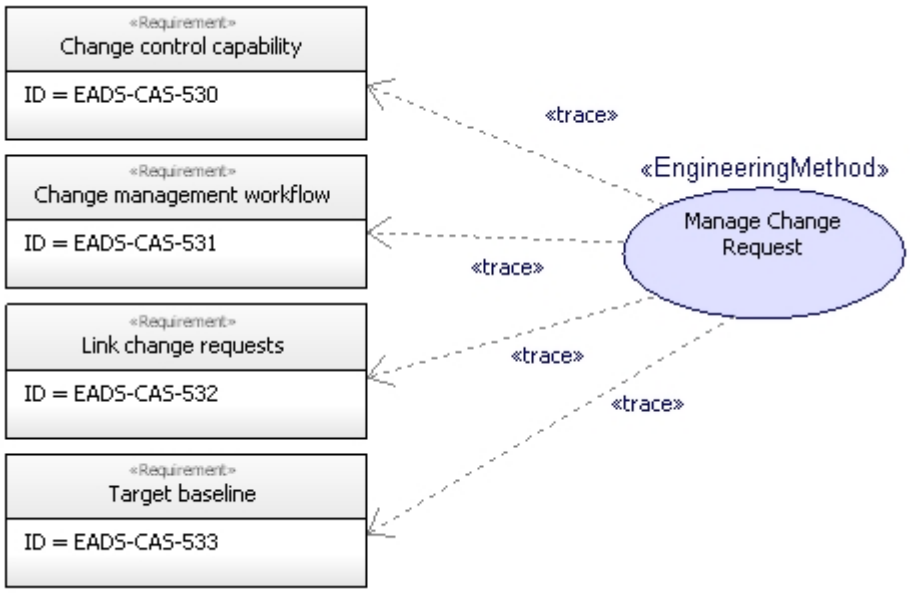

Figure 8-19 EM205_02_01 Manage Change Requests

EADS-CAS-531

The SEE shall provide change management workflow functions. The SEE shall provide the capability to track change requests and to account their status. According to a user-modifiable workflow and depending on the status of the change request, the SEE shall notify persons about their assigned tasks. The SEE shall update the status of the change requests according to the workflow, if a person has completed his tasks.

EADS-CAS-532

The SEE shall allow problem reports and change request to be linked to [Configuration Items] e.g. to requirement modules, design elements or test cases.

EADS-CAS-530

The SEE shall allow to manage Problem Reports, Change Request, Change Proposals and Change Orders.

EADS-CAS-533

The SEE shall allow to define a target baseline, that reflects a functional scope of an envisaged configuration, listing all the changes necessary to be implemented in the current baseline.

Add. Info: PDR, CDR,…

8.8 PA205_03 - User and User Access Management

8.8.1 EM205_03_01 - Manage Users

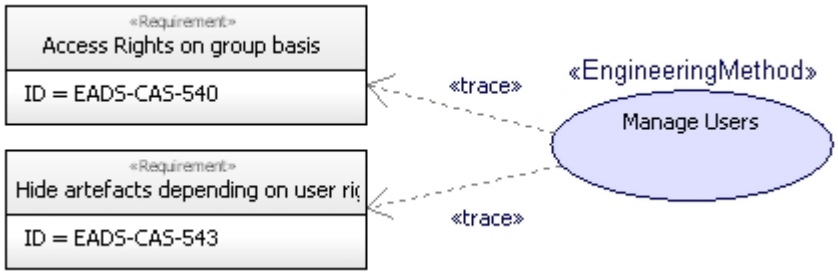

Figure 8-20 EM205_03_01 Manage Users

EADS-CAS-540

The SEE shall control user access rights on group basis.

EADS-CAS-543

The SEE security and access management shall consider restriction levels down to single [Configuration Items]. During export and document generation, only those [Configuration Items] shall be considered where the current user has the appropriate rights.

Add. Info: The group policies are applied down to object level and artefacts so to configure access rights and viewing rights. The rights could be assigned considering need-to-know principles.
E.g. sub-contractors or SW/HW development teams working on a specific part. Especially for sub-contractors the security status need to be considered, e.g. if certain information is "Company Confidential" or relevant for a not yet registered patent.

8.8.2 EM205_03_02 - Load Artefact

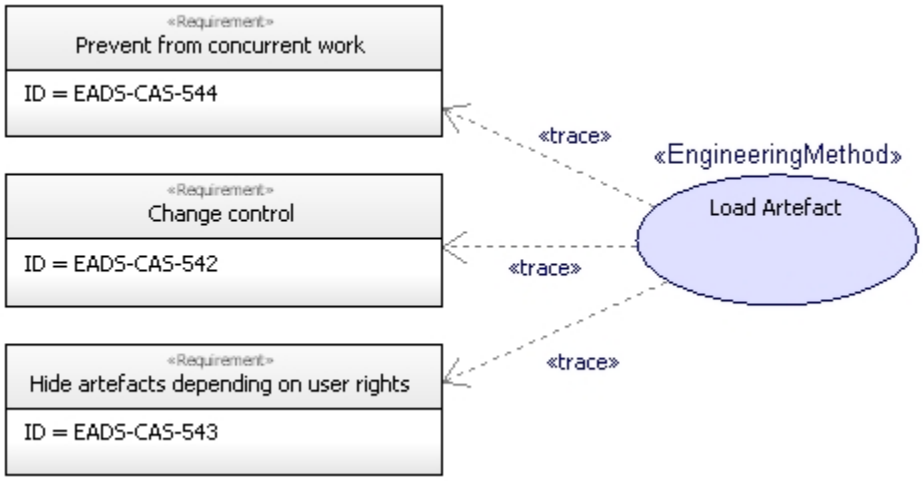

Figure 8-21 EM205_03_02 Load Artefact

EADS-CAS-544

The SEE shall prevent from
- concurrent modification of artefacts by different users and
- data corruption from concurrent work.

Add. Info: Advanced locking mechanism allows the parallel work with minimum restrictions for the user.

EADS-CAS-542

The SEE shall control all identified [Configuration Items] with respect to history, tracing, configuration and change control.

Add. Info: Artefacts include also documents, such as .doc(x), .ppt(x), .xls(x), .jpg, .vsd, ...

EADS-CAS-543

The SEE security and access management shall consider restriction levels down to single [Configuration Items]. During export and document generation, only those [Configuration Items] shall be considered where the current user has the appropriate rights.

Add. Info: The group policies are applied down to object level and artefacts so to configure access rights and viewing rights. The rights could be assigned considering need-to-know principles.
E.g. sub-contractors or SW/HW development teams working on a specific part. Especially for sub-contractors the security status need to be considered, e.g. if certain information is "Company Confidential" or relevant for a not yet registered patent.

8.8.3 EM205_03_03 - Sign On SEE and create New Artefact

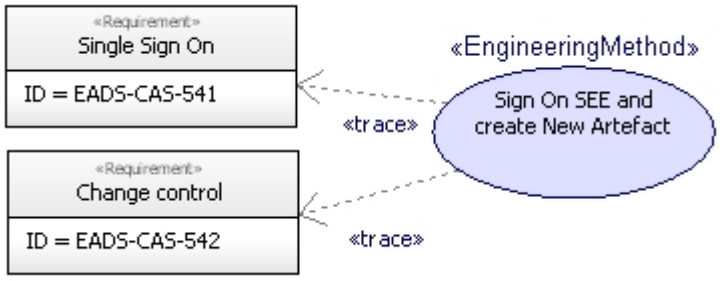

Figure 8-22 EM205_03_03 Sign On SEE and create New Artefact

EADS-CAS-541

The SEE shall provide single sign on capability.

Add. Info: No individual login for each single tool.

EADS-CAS-542

The SEE shall control all identified [Configuration Items] with respect to history, tracing, configuration and change control.

Add. Info: Artefacts include also documents, such as .doc(x), .ppt(x), .xls(x), .jpg, .vsd, ...

8.9 PA205_04 - Process Automation

8.9.1 EM205_04_01 - Generate Tool Inventory

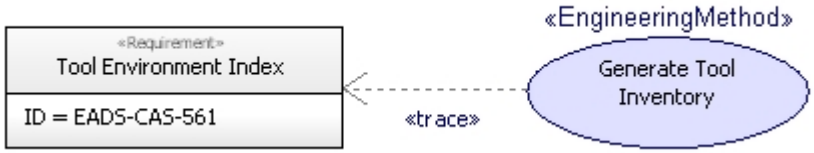

Figure 8-23 EM205_04_01 Generate Tool Inventory

EADS-CAS-561

The SEE shall provide the capability for automatic generation of a tool environment configuration index.

Add. Info: Generation of a SEE tool configuration index including the configuration and revision information of each tool integrated in the SEE.

8.9.2 EM205_04_02 - Save All

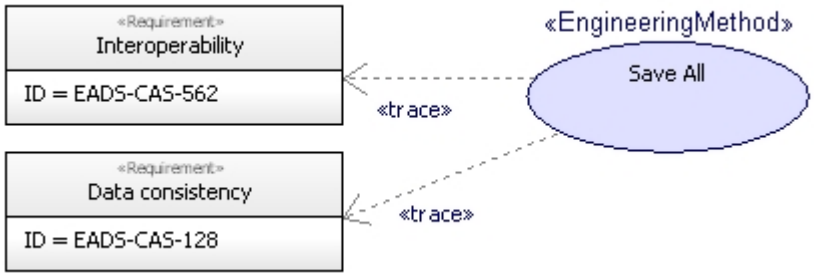

Figure 8-24 EM205_04_02 Save All

EADS-CAS-562

The SEE shall provide interoperability capabilities between the different integrated tools. Tools implementing and exposing similar functions shall be exchangeable and shall provide identical API for automation.

Add. Info: Efficient SE is only possible, when the tools are integrated. This avoids unnecessary data-exchange problems (formats, handling, etc.).

EADS-CAS-128

The SEE shall ensure consistency of the data. Data shall not be duplicated.

Add. Info: Tools are responsible to manage their own data. There shall be a functional cross-tool interface to access the data in real-time. Therefore, the idea is not to import the data from one tool into another tool, and work on that data in the other tool. Of course, for purpose of backup, caching, archiving the data may be duplicated.

8.9.3 EM205_04_03 - Search Data

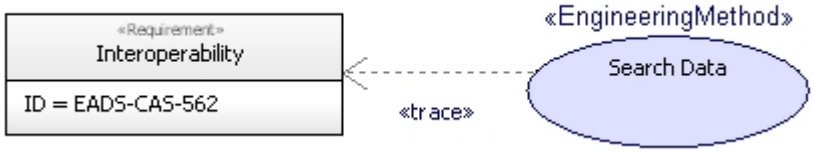

Figure 8-25 EM205_04_03 Search Data

EADS-CAS-562

The SEE shall provide interoperability capabilities between the different integrated tools. Tools implementing and exposing similar functions shall be exchangeable and shall provide identical API for automation.

Add. Info: Efficient SE is only possible, when the tools are integrated. This avoids unnecessary data-exchange problems (formats, handling, etc.).

8.10 PA206_01 - Maintain Traceability

8.10.1 EM206_01_01 - Retrieve Valid Traces

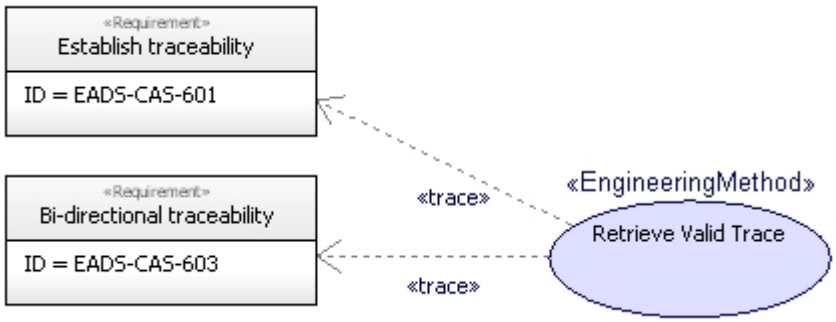

Figure 8-26 EM206_01_01 Retrieve Valid Traces

EADS-CAS-601

The SEE shall provide the capability to link different artefacts of the various tools integrated in the SEE.

Add. Info: [Artefacts] are hosted and exposed by their respective tools. Therefore, [links] shall not be limited to the scope of a single tool. Links shall have a [link source] and a [link target]. The link source has an [outgoing link]. The target has an [incoming link]. [Suspect links] are those links where the target has changed after the source has changed.

EADS-CAS-603

The SEE shall provide the capability to follow [links] in both directions.

Add. Info: In both directions mean from [link source] to [link target] and vice versa from [link target] to [link source].

8.10.2 EM206_01_02 - Analyse Trace

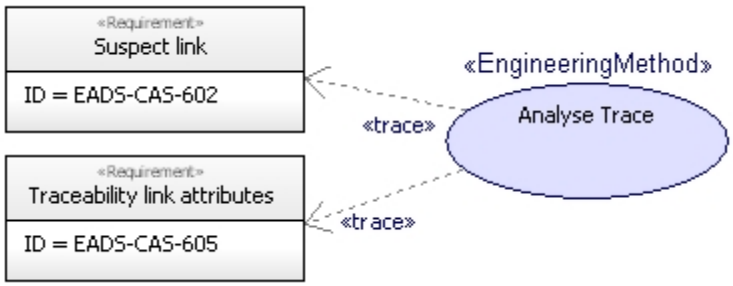

Figure 8-27 EM206_01_02 Analyse Trace

EADS-CAS-602

The SEE shall mark a [link] as suspect, when at least one of the linked artefacts is changed.

EADS-CAS-605

The SEE shall provide following attributes for [links] between artefacts of the SEE:

- type
- source
- target.

8.11 PA206_02 - Monitor Development Progress

8.11.1 EM206_02_01 - Perform Coverage Analysis

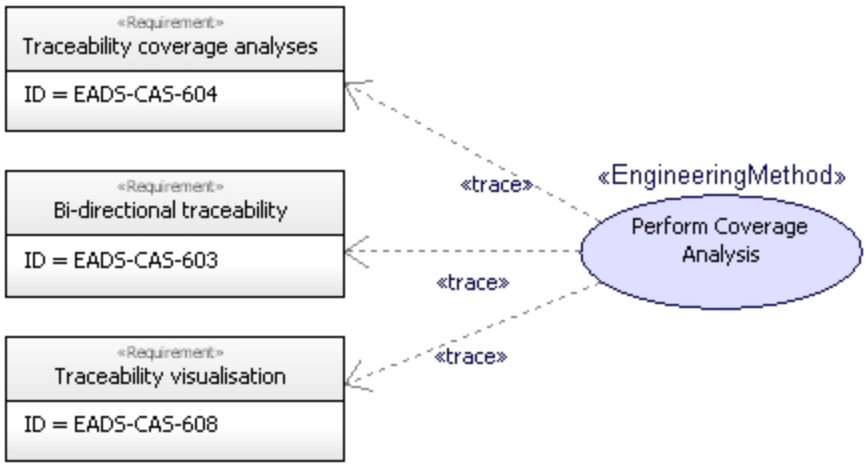

Figure 8-28 EM206_02_01 Perform Coverage Analysis

EADS-CAS-604

The SEE shall provide the capability to perform different [link] based analyses i.e.

- to check for a set of artefacts if all artefacts have an [incoming link] of a dedicated [link type]
- to check for a set of artefacts if all artefacts have an [outgoing link] of a dedicated [link type]
- to check for a set of artefacts if there are [suspect links]

EADS-CAS-603

The SEE shall provide the capability to follow [links] in both directions.

Add. Info: In both directions mean from [link source] to [link target] and vice versa from [link target] to [link source].

EADS-CAS-608

The SEE shall provide the capability to visualise for a selected artefact all the linked artefacts recursively over a user selected number of levels considering the [link type].

Add. Info: For example to support a change impact analysis.

8.12 PA206_03 - Perform Verification

8.12.1 EM206_03_01 - Create Verification Objective

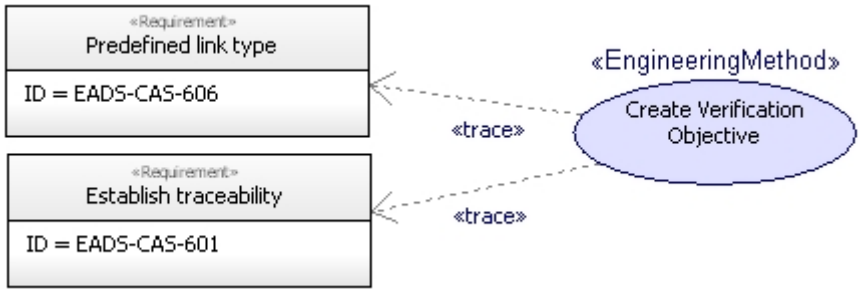

Figure 8-29 EM206_03_01 Create Verification Objective

EADS-CAS-601

The SEE shall provide the capability to link different artefacts of the various tools integrated in the SEE.

Add. Info: [Artefacts] are hosted and exposed by their respective tools. Therefore, [links] shall not be limited to the scope of a single tool. Links shall have a [link source] and a [link target]. The link source has an [outgoing link]. The target has an [incoming link]. [Suspect links] are those links where the target has changed after the source has changed.

EADS-CAS-606

Each [link] within the SEE shall have its own type specifying the usage or meaning of the [link]. The SEE shall support following pre-defined types:

- verify
- satisfy
- derive
- refer
- validate
- modify
- decompose
- refine
- justify
- classify

8.12.2 EM206_03_02 - Create Verification Case

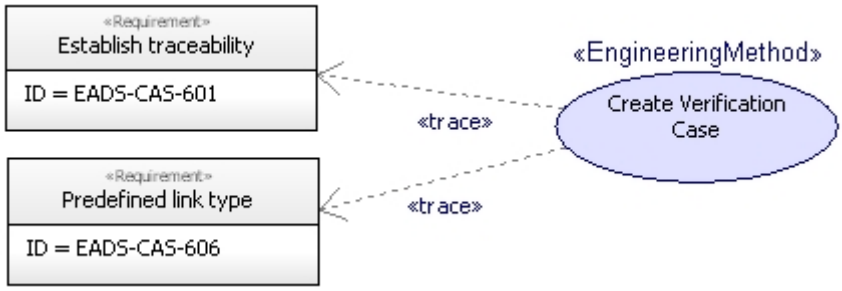

Figure 8-30 EM206_03_02 Create Verification Case

EADS-CAS-601

The SEE shall provide the capability to link different artefacts of the various tools integrated in the SEE.

Add. Info: [Artefacts] are hosted and exposed by their respective tools. Therefore, [links] shall not be limited to the scope of a single tool. Links shall have a [link source] and a [link target]. The link source has an [outgoing link]. The target has an [incoming link]. [Suspect links] are those links where the target has changed after the source has changed.

EADS-CAS-606

Each [link] within the SEE shall have its own type specifying the usage or meaning of the [link]. The SEE shall support following pre-defined types:

- verify
- satisfy
- derive
- refer
- validate
- modify
- decompose
- refine
- justify
- classify

8.12.3 EM206_03_03 - Create Verification Procedure

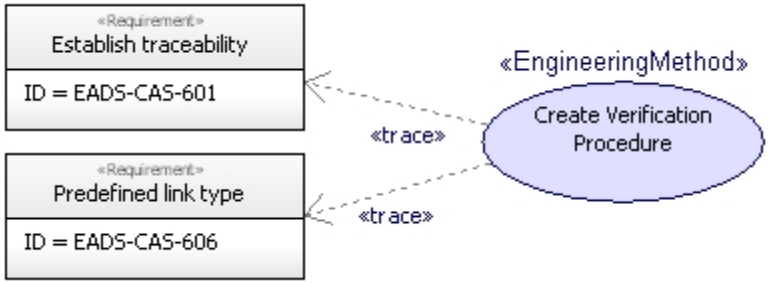

Figure 8-31 EM206_03_03 Create Verification Procedure

EADS-CAS-601

The SEE shall provide the capability to link different artefacts of the various tools integrated in the SEE.

Add. Info: [Artefacts] are hosted and exposed by their respective tools. Therefore, [links] shall not be limited to the scope of a single tool. Links shall have a [link source] and a [link target]. The link source has an [outgoing link]. The target has an [incoming link]. [Suspect links] are those links where the target has changed after the source has changed.

EADS-CAS-606

Each [link] within the SEE shall have its own type specifying the usage or meaning of the [link]. The SEE shall support following pre-defined types:

- verify
- satisfy
- derive
- refer
- validate
- modify
- decompose
- refine
- justify
- classify

8.13 PA206_04 - Analyse Impact of Change

8.13.1 EM206_04_01 - Analyse Impact of Change

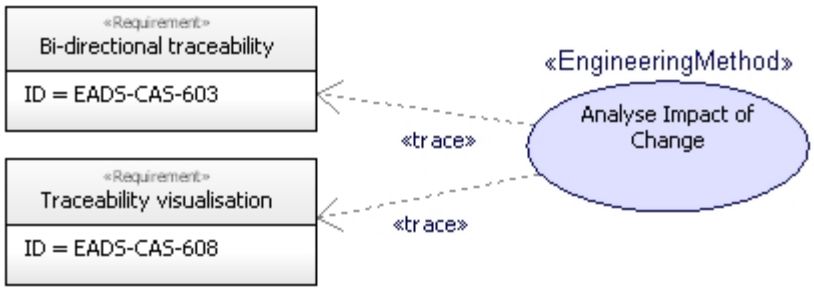

Figure 8-32 EM206_04_01 Analyse Impact of Change

EADS-CAS-608

The SEE shall provide the capability to visualise for a selected artefact all the linked artefacts recursively over a user selected number of levels considering the [link type].

Add. Info: For example to support a change impact analysis.

EADS-CAS-603

The SEE shall provide the capability to follow [links] in both directions.

Add. Info: In both directions mean from [link source] to [link target] and vice versa from [link target] to [link source].