 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

IOS design requirements

D503.030

DOCUMENT INFORMATION

Project	CRYSTAL
Grant Agreement No.	ARTEMIS-2012-1-332830
Deliverable Title	IOS design requirements
Deliverable No.	D503.030
Dissemination Level	CO
Nature	R
Document Version	V1.0
Date	2014-04-30
Contact	Pascal POISSON
Organization	ALSTOM
Phone	
E-Mail	pascal.poisson@transport.alstom.com

AUTHORS TABLE

Name	Company	E-Mail
Vidal Delmas TCHAPET NYA	ALSTOM	vidal-delmas.tchapet-nya-ext@transport.alstom.com

REVIEW TABLE

Version	Date	Reviewer
Internal Review	2014-04-14	Pascal POISSON
Internal Review	2014-04-14	Elie Soubiran
Internal Review	2014-04-14	Fateh GUENAB
External Review	2014-04-25	Frédérique VALLÉE
External Review	2014-04-25	Alexandre GINISTY

CHANGE HISTORY

Version	Date	Reason for Change	Pages Affected

CONTENT

Content of Figures

Content of Tables

1 Introduction

1.1 Role of deliverable

The aim of this document is to provide IOS design requirements. The IOS design requirements are generated from the deliverable IOS needs for RTP specification. The IOS design requirements derive from the interoperability between the requirements analysis, the system analysis and the safety analysis. The IOS design requirements also depend on the interoperability between the requirements management tool, the modelling tool and the safety analysis tool used in Alstom Transport System Life Cycle.

1.2 Relationship to other CRYSTAL Documents

The document is related to following deliverables:

- D503.010 - Use case definition: This deliverable describes the case studied in the rail domain. In this deliverable, we have described the system and the system functions. The IOS design requirements are defined from the system breakdown structure.
- D503.020 - IOS needs for RTP specification: The IOS design requirements are generated from the IOS needs provided by this deliverable.
- D604.011 - Specification, Development and Assessment for Safety Engineering: This deliverable focuses on the tools supporting the IOS design requirements.

1.3 Structure of this document

This document is organized as follows:

- Chapter 2 reminds the Use Case: Communication Based Train Control system
- Chapter 3 focuses on the IOS design requirements Overview
- Chapter 4 deals with the IOS design requirements for MBSE
- Chapter 5 deals with the IOS design requirements for MBSA
- Chapter 6 Terms, Abbreviations and Definitions
- Chapter 7 References
- Chapter 8 Annex

2 Use Case: Communication Based Train Control system

The system studied is the Communication Based Train Control system (CBTC). The CBTC is a railway signalling system that makes use of the telecommunications between the train and track equipment for the traffic management and the infrastructure control. The CBTC partial component is illustrated in the Figure 5. The figure shows two CBTC Subsystems, their functions and their Sub functions.

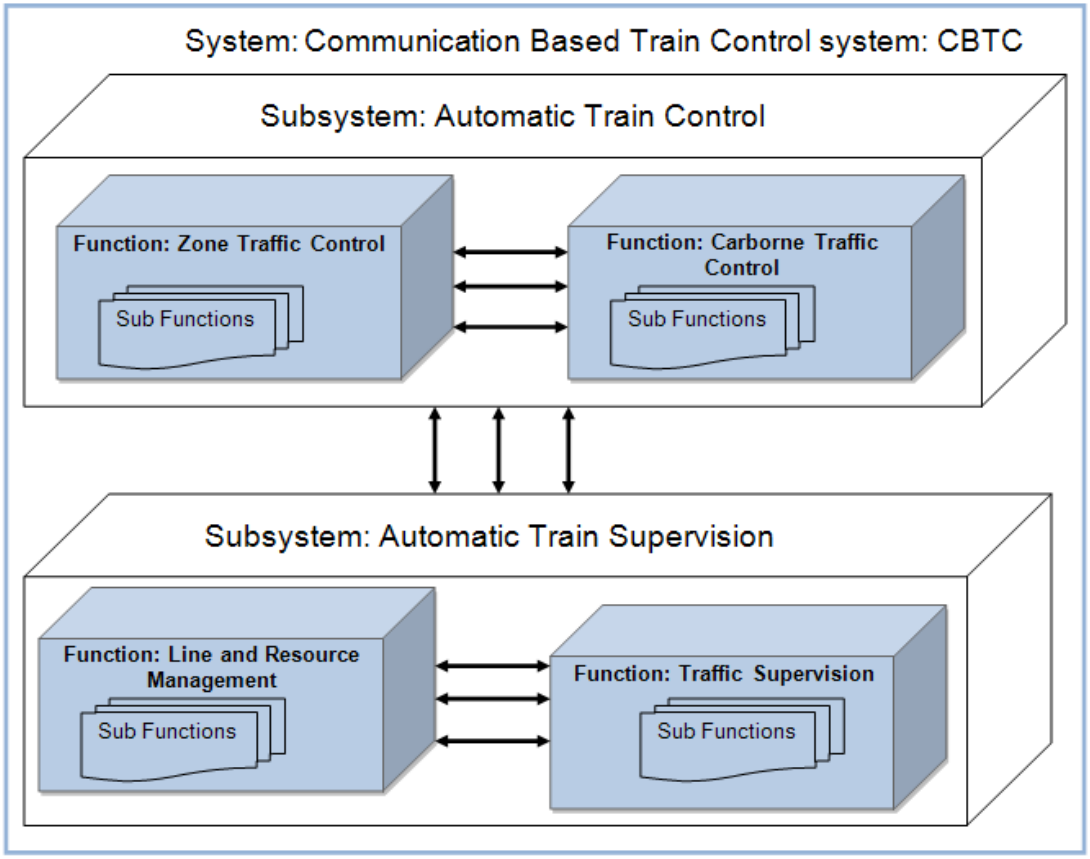

Figure 5 Generic System and interoperability needs

The Alstom Transport Use case focuses on the Carborne Traffic Control Function. The Carbone Traffic control encompasses two main functions: The Ensure Train Protection Function and the Ensure train Operation Function. The analysis also takes into account the sub functions of both functions.

We need to explore the requirement data model for the system. The requirement Data Model is described at four levels:

- Data Model Level 1 - Project Requirements : This level describes the user and project requirements

- Data Model Level 2 - System Specification Level: This level describes the system missions and the functions that satisfy the project requirements.

    - The system Specification encompasses the System Operational Specification (SyOS), the System Functional Specification (SyFS), The System Interface Description (sylD), and the System Architecture Description (SyAD). The SyOS describes system missions

and scenarios by using SysML use cases diagrams for “Mission” and Sequence Diagrams for scenarios. The SyFS describes the system functions and the functional requirements by using SysML Requirements diagrams. The SyID describes the system interface requirement and traces to “function allocated to sub-system”. The SyAD describes the system architecture and traces to system mission or functions.

- The System Traceability matrix allows managing traceability.

From the system specification level, we derive the requirements that the tools must fulfil. These requirements are defined during the activities related to the Model Based System Engineering.

- Data Model Level 3 (Sub System): This level encompasses the Sub-System Specification Data Model and the equipment (Framework). The Sub system Data Model contains requirements to describe in more details the system requirements model.

The third level provides the requirements that must be addressed in the Model Based System Engineering section.

- Data Model Level 4 - Product (Software, Hardware) : The product level is made of two kinds of specifications:
  - A Software/Hardware specification described within DOORS and allocated to the physical product.
  - The Software specifications described out of DOORS. Those Software Specifications are imported within DOORS in order to generate traceability matrix.

The next section of this deliverable deals with the IOS design requirements overview. We give an illustration of Model Based System Engineering and the Model Based Safety Analysis. This section includes relationships between system Life Cycle activities, the requirements, the traceability and the interoperability.

3 IOS design requirements overview

The aim of this section is to give an overview of the Interoperability specification design requirements from the Alstom Transport point of view. The interoperability indicates how readily the system can exchange data and services with other software systems and how easily it can be integrated with external hardware devices [KWSOREQ].

The IOS design requirements are much closed to the traceability links between the artefacts identified during the System Life Cycle global process. We are going to derive IOS Design requirements from the traceability relationships identified from the global system Life Cycle.

Traceability is the extent to which products of each phase can be traced back to products of previous phases. Requirements traceability is used to ensure that each software requirement has been designed and implemented. Each requirement is traced to the software architecture and to the implemented code modules. Requirements traceability tables are a useful tool during software architecture reviews for analysing whether the software architecture has addressed all the software requirements [HGomaa]. Traceability matrices show where entities in the system are defined and referenced.

In this deliverable traceability depends on the Alstom Transport system life cycle and the tools involved in the use case. Those tools are the requirements management tool (DOORS/ReqIF), the modelling tool (Papyrus) and the safety analysis tool (Safety Architect). The System Life Cycle includes beyond others system engineering activities and safety analysis. The system analysis encompasses Operational View, Functional View, and constructional View. It is supported by the SysML modelling standard. The Safety Analyses are made with the Preliminary Hazard Analysis, the System Hazard Analysis and the Sub-Systems Hazard Analysis.

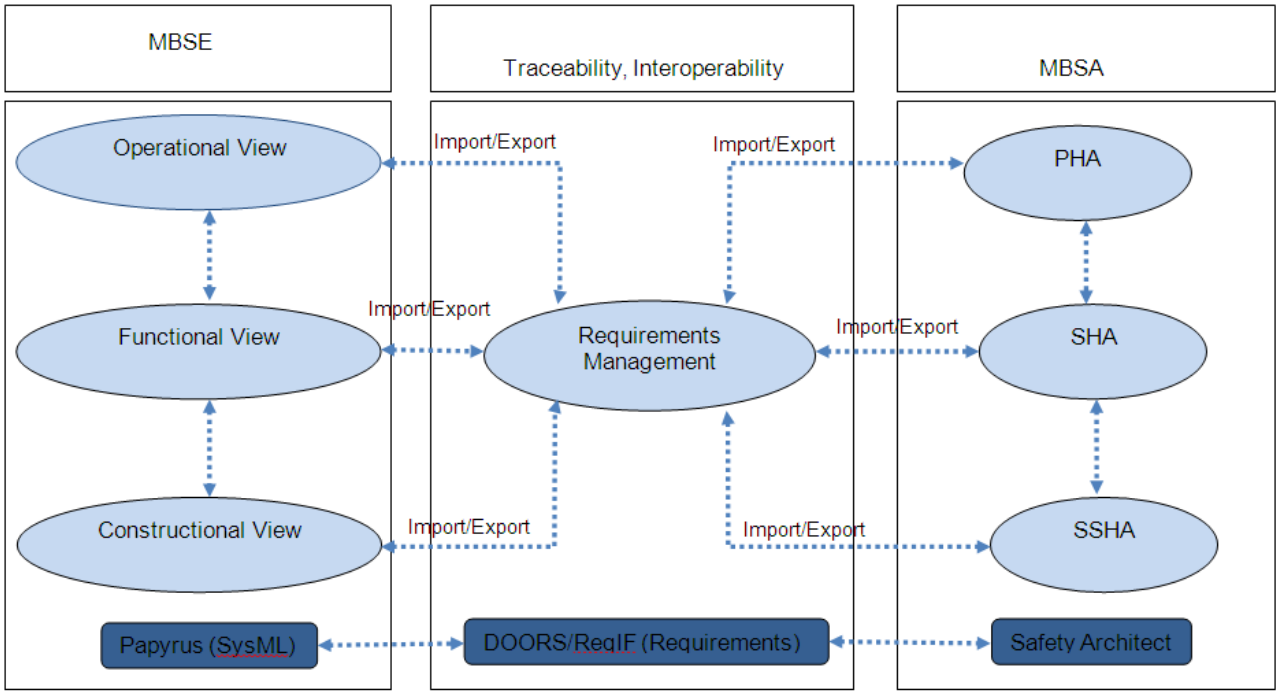

Figure 6 IOS Design Requirements Overview

The safety analysis is run by Safety Architect tool from All4Tec Company. Fault Tree Analysis and Failure Mode Effect Analysis are derived from the Safety models. System Analysis and Safety Analysis interoperate via the requirements management. DOORS and ReqIF are successively used in order to manage the requirements and to exchange data from one tool to another. The IOS design requirements in the global process is depicted in the Figure 6. The user can import/Export data from one tool to another tool.

The next section deals with the Model Based System Engineering. We are going to describe the main activities related to this process. We are going to address the IOS design requirements related to the System Analysis.

4 IOS design requirements for MBSE

The stakeholders involved in this process are the Requirements Engineers, the System Engineers and the Verifier. The First is in charge of managing requirements. The requirements Engineers identify, check, allocate, trace and maintain requirements. The System Engineers are in charge of defining, assigning and monitoring system development activities in order to meet customer's quality, safety, cost and delay requirements. The Verifier is in charge of checking traceability and requirements correctness according to a verification checklist.

The IOS design requirements are derived from:

- The scenarios or activities performed by the stakeholders involved in the system life cycle
- The way the data are exchanged between the tools used in the global process

4.1 Requirements Engineering Activities

This sub section deals with the main activities performed during the Model Based System Engineering. We have identified the following activities:

- Requirements collection and identification
- Requirements clarification
- Requirements analysis
- Negotiation commitment with receiver
- Requirements delivery & configuration management
- Requirements development and trace
- Requirements verification and validation
- Requirements change management

Each activity is described in a table. The Table contains the purpose, the inputs required to perform each activity, the task allocated to the stakeholder in charge of the activity, the outputs provided by each activity, and the tools that support each of them.

The activities related to the Model Based System Engineering are described in the next paragraphs.

4.1.1 Requirements collection and identification activities

Purpose	Propose or Capture and identify requirements in order to constitute a repository of identified requirements
Input work products	Source documents
	Requirements management plan

Tasks	Requirement Engineer: - Captures each requirement statement in source document. A unique requirement identifier is generated automatically. - Sets requirements attributes according to the previous information (Name, Type, Definition, Source)
Output work products	Captured requirements in user needs module with Status = Defined
Supporting tools	DOORS

Table 1 Requirements collection and identification activities

From this activity, the following requirements are identified:

R-WP3/D503.030-1 Requirements shall trace the requirements source document

R-WP3/D503.030-2 Requirements shall be compliant to the requirements management plan

The requirement source includes the technical reference materials, the customers' needs, the standards, the rules, the stakeholders meeting reports and so on.

4.1.2 Requirements clarification activities

Purpose	Refine each requirement to reach a shared understanding between requirement provider and requirement receiver.
Input work products	Requirements in user needs module with Status = Defined
Tasks	Requirement receiver and requirement provider jointly clarify requirements. This clarification can lead to re-formulate the requirement or create additional requirements. Requirement provider and requirement receiver finally reach a common understanding on the requirement.
Output work products	Agreed requirements in user needs module with Status=Committed
Supporting tools	DOORS

Table 2 Requirements clarification activities

From this activity, the following IOS design requirements are identified:

R-WP3/D503.030-3 The traceability links between provided and received requirements shall be established. The person that provides the requirements must be identified. And the person that receives the provided requirements must be also identified.

R-WP3/D503.030-4 If the requirement changes, a traceability link shall be established between the previous and the current requirements.

4.1.3 Requirements analysis

This activity encompasses the requirements refinement and the commitment negotiation with the receiver.

4.1.3.1 Refine requirements activities

Purpose	Create and refine requirements within a same module to develop the design
Input work products	User need module Requirement module Architecture module
Tasks	Requirements Engineer: - Refines a requirement within a given requirement module - Refines a requirement into new requirements iteratively as many times as necessary till it becomes possible to allocate each refined requirement onto a single component of the system architecture. - Substantiates the refinement design choices filling the attribute - Allocates requirement onto a single component of the system architecture - Sets the requirements Status to Status=defined.
Output work products	Requirements in requirement module, allocated on a single component of the architecture module architecture, with Status = defined
Supporting tools	DOORS, SysML Modelling Tool

Table 3 Refine requirements activities

This activity generates the following IOS design requirements:

R-WP3/D503.030-5 When a requirement is refined, a traceability link shall be established between requirements and refined requirements.

R-WP3/D503.030-6 The traceability links shall be established between the requirements and the single component of the system architecture

R-WP3/D503.030-7 The traceability links shall be established between the SyOS (System Operational Specification) and the project specification

R-WP3/D503.030-8 The traceability links shall be established between the SyFS (System Functional Specification) and the SyOS (System Operational Specification)

R-WP3/D503.030-9 The traceability links shall be established between the SyFS (System Functional Specification) and the project specification

R-WP3/D503.030-10 The traceability links shall be established between the SyID (System Interface Description) and the function allocated to sub-system

R-WP3/D503.030-11 The traceability between the SyAD (System Architecture Description) and the system functions shall be established

R-WP3/D503.030-12 The product Requirements shall trace the System Requirements

4.1.3.2 Negotiating the commitment with receiver

Purpose	Negotiate with requirement receiver to obtain common understanding on the requirement and commitment from requirement receiver
Input work products	Requirements in requirement module with < Status> =defined Architecture module
Tasks	Requirement receiver and requirement provider jointly clarify requirements to reach a common understanding of the requirement. This clarification can lead to re-formulate the requirements or create additional requirements. Requirement provider and requirement receiver finally reach a common understanding on the requirement < Status> is set to <Status> committed.
Output work products	Requirements in requirement module, allocated on a single component of the architecture module architecture, with <Status> =committed
Supporting tools	DOORS, SysML Modelling Tool

Table 4 Negotiating commitment with receiver

From this activity, the following requirement is derived:

R-WP3/D503.030-13 The following requirements shall be addressed: **R-WP3/D503.030-3, R-WP3/D503.030-4**

4.1.4 Requirement delivery & configuration management activities

Purpose	Deliver a given version of requirements to enable work at sub-system level on a frozen reference
Input work products	Requirements in requirement module, allocated on a single component of the architecture module architecture, with <Status> =committed
Tasks	Requirements Engineer: - Delivers the documentation views related to all committed requirements - Status is set to <Status> = approved for committed requirements. Status is unchanged for obsolete requirements
Output work products	Requirements with <Status> = approved or obsolete Documentation
Supporting tools	DOORS

Table 5 Requirement delivery & configuration management activities

This activity generates the following requirements:

R-WP3/D503.030-14 Traceability relationships between the documentation view and the committed requirements shall be provided

R-WP3/D503.030-15 Traceability relationships between the requirements shall be provided

4.1.5 Develop and trace requirements

Purpose	Develop the requirements allocated to a sub-system and trace the requirement to the system requirement
Input work products	Requirements with Status = approved
Tasks	System Engineer: - Develops the system requirements allocated to the sub-system - Traces the top level sub-system requirements to the system requirements allocated to the sub-system. - Updates traceability links that were linked to requirements with Status=obsolete
Output work products	Sub-system requirements
Supporting tools	DOORS

Table 6 Develop and trace requirements

This activity generates the following IOS design requirements:

R-WP3/D503.030-16 Traceability matrix between the top level sub-system requirements and the system requirements allocated to the sub-system shall be provided

R-WP3/D503.030-17 The traceability links between requirements status shall be provided. If the requirements status changes, the modification must be provided

R-WP3/D503.030-18 The Subsystem Requirements shall trace the Subsystem functions

R-WP3/D503.030-19 The product Requirements shall trace the Subsystem Requirements

4.1.6 Requirements verification and validation activities

This activity intends to verify the requirements characterization and to verify requirements traceability.

4.1.6.1 Verify requirements characterization

Purpose	Verify that requirement statement is defined according to verification rules and follows System modelling rules
Input work products	Requirements, with Status = committed or approved
Tasks	Verifier performs the activities defined in System V&V plan as briefly reminded hereunder: - Verifies that requirement statement follows System verification rules, - Verifies that requirement statement follows System modelling rules, - Verifies that requirement characterization follows System requirement management plan - Verifies that agreed requirements are allocated on a single component of the system architecture, - Updates the <Characterization verification> and <Characterization verification rationale> attributes. In case of discrepancy detected on a requirement with Status = approved, the Verifier logs a change request. In case of discrepancy detected on requirement with Status = committed, verifier requests System Engineer to update the requirement.
Output work products	Requirements with updated <Characterization verification> and <Characterization verification rationale> attributes Change request
Supporting tools	DOORS, SysML Modelling Tool,

Table 7 Verify requirements characterization

This activity generates the following IOS design requirements:

R-WP3/D503.030-20 Traceability between the requirement statement and the system verification rule shall be provided

R-WP3/D503.030-21 Traceability between the requirement statement and the system modelling rules shall be provided

R-WP3/D503.030-22 Traceability between the requirement characterization and the system requirements management plan shall be provided

4.1.6.2 Verify requirements traceability

Purpose	Verify requirements traceability
Input work products	Parent module requirements with Status=approved Child module requirements with Status=approved or committed
Tasks	The Verifier performs the activities defined in System V&V plan as briefly reminded hereunder: - Verifies that each requirement in parent module is covered by at least one requirement in a child module, and that the traceability is either obvious or correctly substantiated in the <rationale> attribute of the requirement in child module. - Updates the <Traceability Verification> attribute of the child module requirement In case of discrepancy detected when the requirement in child module is with Status=committed, the Verifier requests System Engineer to update the requirement in child module. In case of discrepancy detected when the requirement in child module is with Status=approved, the Verifier logs a change request
Output work products	Requirements with updated <Traceability Verification> attributes Change request
Supporting tools	DOORS, SysML Modelling Tool,

Table 8 Verify requirements traceability activities

For this activity, the following IOS requirements shall be fulfilled:

R-WP3/D503.030-23 Each requirement in parent module shall cover at least one requirement in a child module

R-WP3/D503.030-24 The traceability verification attribute of the child module requirement shall be updated

4.1.7 Requirements change management activities

Purpose	Implement changes
Input work products	Requirements, with Status = Approved Analysed Change request
Tasks	The Requirements Engineer implements changes required in the change order (i.e. analysed change request): - In case a requirement is modified: set requirement Status to obsolete, and duplicates the requirement in a new object on which the modification is applied - In case a requirement is deleted: set requirement Status to obsolete. - Since the purpose of the change control management is to reach agreement of the stakeholder, the Status of the newly created requirements can be set to Committed

Output work products	Requirements, with Status = Obsolete or Committed
Supporting tools	DOORS, SysML Modelling Tool,

For this activity, the following IOS requirements shall be fulfilled:

R-WP3/D503.030-25 The following requirements shall be fulfilled: **R-WP3/D503.030-4**

The next sub section explains the way data are exchanged when performing the Model Based System Engineering.

4.2 Data exchange between the Requirements management tool and the Modelling tool

This section deals with the tools used during the Model Based System engineering process and the way data are exchanged between tools. Model Based System Engineering requires and the Requirements management tool DOORS to manage requirements, the Requirements Interchange Format tool ReqIF in order to ensure data exchange between the requirements tool and the Table 9 Requirements change management activities modelling tool.

The data Exchange between DOORS and the modelling Tool is described in the Table 10 .

PURPOSE	The purpose of this activity is to show the way data are exchanged between the requirements tool and the modelling tool.
INPUTS	System requirements
OUTPUTS	Refined system requirements
DESCRIPTION	The System Engineer: • Imports requirements from the DOORS format to the ReqIF format • Imports requirements from the ReqIF format to the Modelling Tool format • Performs System Analysis • Exports refined requirements from the Modelling Tool format to the ReqIF format • Exports refined requirements from the ReqIF format to the DOORS format
TOOLS	DOORS, Modelling Tool

Table 10 Data exchange scenario between DOORS and the modelling Tool

The Figure 7 gives an illustration of the data exchanged between the requirements tool and the Modelling tool via the requirements interchange format ReqIF.

Figure 7 Data Exchanged between the requirements management tool and the modelling tool via the requirements interchange format ReqIF

Here, we derive requirements related to the data exchanged between the requirements and the modelling tools.

R-WP3/D503.030-26 The System Engineer shall be able to import the System/Subsystem requirements from the DOORS format to the ReqIF format

R-WP3/D503.030-27 The System Engineer shall be able to import the System/Subsystem requirements from the ReqIF format to the modelling tool format

R-WP3/D503.030-28 The System Engineer shall be able to export the refined requirements from the modelling tool format to the ReqIF format

R-WP3/D503.030-29 The System Engineer shall be able to export the refined requirements from the ReqIF format to the DOORS format

R-WP3/D503.030-30 During the import/export phase, the hierarchical description of the system components and the system functions shall be preserved.

R-WP3/D503.030-31 During the import/export phase, the integrity of the data shall be preserved

R-WP3/D503.030-32 The transformation from the DOORS format to the ReqIF format shall be a hundred percent succeeded

R-WP3/D503.030-33 The transformation from the ReqIF format to the DOORS format shall be a hundred percent succeeded

R-WP3/D503.030-34 The transformation from the ReqIF format to the Modelling tool format shall be a hundred percent succeeded

R-WP3/D503.030-35 The transformation from the Modelling tool format to the ReqIF format shall be a hundred percent succeeded

R-WP3/D503.030-36 The data exchanged between the tools shall be secured

R-WP3/D503.030-37 The data exchange between ReqIF, DOORS and the modelling tool shall be quickly performed. This requirement depends on the data size that we have to import/export.

The next section deals with the IOS design requirements related to the Model Based Safety Analysis. In this section we are going to define the IOS design requirements from the Preliminary Hazard Analysis, the System Hazard Analysis, the Subsystem Hazard Analysis and the Hazard Log.

5 IOS design requirements for MBSA

5.1 Model Based Safety Analysis Activities

This section gives an overview of the safety activities of the System Life Cycle. For this deliverable, we focus on the main safety activities such as the:

- Preliminary Hazard Analysis
- System Hazard Analysis
- Subsystem Hazard Analysis
- Hazard Log

For each activity, we give: its purpose, its inputs, its outputs, its description, and the tools that support it. The IOS design requirements derive from the scenarios described for each safety activity.

5.1.1 Preliminary Hazard Analysis Activities

The preliminary hazard analysis (PHA) technique is a safety analysis activity for identifying hazards, their associated causal factors, effects, levels of risk, and the mitigating design measures. It provides the way for identifying and collecting hazard. The initial safety requirements are derived from the PHA.

The preliminary Hazard Analysis scenario is shown in the Table 11. The basic inputs for PHA include the Safety Plan, the System User Needs, and the Hazard Breakdown Structure. The PHA provides Hazards, Hazards cause, Safety requirements for eliminating and mitigating Hazards.

PURPOSE	Define the high level requirements to be applied with the design or to be exported in order to cover the hazards identified in the Hazard Breakdown Structure (HBS).
INPUTS	Safety Plan User Needs Hazard Breakdown Structure
OUTPUTS	Preliminary Hazard Analysis
DESCRIPTION	The team in charge of this activity will: • Identify protections necessary to eliminate or mitigate identified risks which will have to be demonstrated.
TOOLS	DOORS

Table 11 Scenarios related to the Preliminary Hazard Analysis

The requirements related to the Preliminary Hazard Analysis are defined as follow:

R-WP3/D503-38 The System requirements becoming safety requirements shall be traced
R-WP3/D503-39 The traceability between potential accidents and the related Hazard shall be provided
R-WP3/D503-40 The allocated Tolerable Hazard Rate (THR) to each Hazardous situation shall be provided
R-WP3/D503-41 The traceability matrix of all potential Hazards to Hazard causes shall be established

R-WP3/D503-42 The traceability Relationships between the safety requirements and the potential hazards shall be provided

R-WP3/D503-43 The requirements allocated to the systems shall be provided

R-WP3/D503-44 All The safety requirements shall be compliant to the safety plan document

R-WP3/D503-45 At the end of the PHA, the hazard log shall be updated with the Hazards and the related safety requirements

5.1.2 System Hazard Analysis Activities

The System hazard Analysis (SHA) is the process used for evaluating risk and the safety compliance at the system level. This methodology focuses on the functions of the system. During the SHA process, the Safety Engineer makes sure that Hazards causes are identified and mitigated. He also verifies whether the overall system risk is identified and accepted. When performing System Hazard Analysis, the following factors should be taken into account: The Safety Plan, the Preliminary Hazard Analysis, the Hazard Breakdown Structure, the System Functional Specification, the System Operational and Support Hazard Analysis.

PURPOSE	Analyse the causes of unsafe situations of the system related with the functions it implements. And define the actions and the means that eliminate, or reduce to an acceptable level, the risks.
INPUTS	Safety Plan Preliminary Hazard Analysis Hazard breakdown structure System Functional Specification System Operational and Support Hazard Analysis
OUTPUTS	System Hazard Analysis
DESCRIPTION	The team in charge of this activity will: • Identify all failures leading to potential hazards through a Failure Mode and Effects Analysis (FMEA). • Determine and assign the Safety Integrity Level (SIL) of the system functions. • Identify barriers and safety requirements against hazardous situations. • Identify the necessary sub-system hazard analyses, specific hazard analyses and interface hazard analyses and record this information in the Hazard Log. • Record identified hazards in the Hazard Log.
TOOLS	DOORS, Safety Architect

Table 12 Scenarios related to the System Hazard Analysis

The following requirements are derived from the System Hazard Analysis.

R-WP3/D503-46 The traceability between the safety requirements and the system requirements shall be provided

R-WP3/D503-47 The failure mode shall trace the related function

R-WP3/D503-48 The traceability matrix of all failure modes to failure causes shall be provided

R-WP3/D503-49 The traceability matrix of all failure modes to identified accident shall be provided

R-WP3/D503-50 The traceability between the failure causes and the failure effects shall be established

R-WP3/D503-51 The traceability between the safety requirements and the related barrier shall be provided

R-WP3/D503-52 The traceability between the system functions and the failure modes shall be provided

R-WP3/D503-53 The SIL allocation to each system function shall be provided

R-WP3/D503-54 The traceability between the safety requirements and the failure modes shall be established

R-WP3/D503-55 The traceability between the safety requirements and the system requirements in order to update and improve system requirements shall be provided

R-WP3/D503-56 All the safety requirements shall be compliant to the safety plan document

R-WP3/D503-57 The Hazard Log shall be updated with the Hazards and the related safety requirements

5.1.3 Subsystem Hazard Analysis Activities

The Subsystem Hazard Analysis (SSHA) is the safety analysis activity for identifying hazards, their associated causal factors, effects, level of risk, and for defining the safety requirements for mitigating and eliminating those hazards. The SSHA is performed when the System Hazard Analysis is available.

PURPOSE	The purpose of this activity is to analyse the causes of failures of the sub-systems, and to define the means to eliminate or reduce them to an acceptable level the risks.
INPUTS	Safety Plan System Hazard analysis System Interface Hazard analysis Sub-systems requirements specification
OUTPUTS	sub-system Hazard analysis
DESCRIPTION	The team in charge of this activity will: • Identify all failures leading to potential hazards through a Failure Mode and Effects Analysis (FMEA). • Determine and assign the SIL of sub-systems functions • Identify barriers and safety requirements against hazardous situations.
TOOLS	DOORS, Safety Architect

Table 13 Scenarios related to the Subsystem Hazard Analysis

The following IOS design requirements are those identified during SSHA:

R-WP3/D503-58 The safety requirements allocation to Sub-Systems shall be provided

R-WP3/D503-59 The traceability between the Sub-System functions and the failure modes shall be provided

R-WP3/D503-60 The SIL shall be allocated to each subsystem function. The aim of this allocation is to decide whether we should apply the safety analysis to this function.

R-WP3/D503-61 The traceability between the safety requirements and the failure modes identified shall be provided

R-WP3/D503-62 The traceability Relationships between the safety requirements and the system requirements in order to update and improve system requirements

R-WP3/D503-63 The traceability matrix of all failure modes to failure causes shall be established

R-WP3/D503-64 The traceability matrix of all failure causes to failure effects shall be established

R-WP3/D503-65 All the safety requirements shall be compliant to the safety plan document

R-WP3/D503-66 The Hazard Log shall be updated with the Hazards and the related safety requirements

R-WP3/D503-67 The Safety Engineer shall be able to generate the FMEA results within an excel Sheet

5.1.4 Hazard Log Activities

Hazard Log is the methodology for evaluating the safety requirements. It is used to ensure that every hazard identified has at least one related safety requirement. It is also in charge of verifying whether all requirements have been implemented and validated. Table 14 illustrates the basic inputs required for performing the Hazard Log, the scenario executing when performing the Hazard Log, and the Outputs provided by the Hazard Log activities.

PURPOSE	The purpose of this activity is to record and give the status of safety requirements
INPUTS	Hazard Breakdown Structure System Preliminary Hazard Analysis System Requirements Specification System and sub-system Requirements Test Plans System and sub-system Requirements Test Descriptions System and sub-system Integration Test Descriptions System and sub-system Requirements Test Reports Operational and Support Hazard Analysis System Hazard Analysis Sub-system Hazard analyses Specific Safety Studies Fault Tree Analysis Products and Software exported constraints
OUTPUTS	System Hazard Log Hazard Log

DESCRIPTION	The aim of this activity is to:
	• Record for each identified hazard the following attributes:
	  ○ An identification number,
	  ○ A complete description,
	  ○ Its consequences,
	  ○ Its estimated frequency,
	  ○ The components it involves,
	  ○ The protections,
	  ○ The associated actions,
	  ○ Its status (open, resolved, closed),
	  ○ The related safety requirements,
	• Record people involved in safety related activities with their skills;
	• Record methods, techniques and tools used for analysis;
	• Record hypothesis used for analysis;
	• Record known limits of analysis;
	• Record level of confidence on used data for analysis.
	• Verify the coverage of safety requirements by tests cases

TOOLS	DOORS, Safety Architect

Table 14 Scenarios related to the Hazard Log

From the Hazard Log activities the following IOS design requirements are defined.

R-WP3/D503-68 The safety analysis tool shall allow to allocate attribute for each hazard identified

R-WP3/D503-69 The safety tool shall allow to allocate activity to the author of the safety requirements

R-WP3/D503-70 The tool safety tool shall allow to record methods, techniques and tools used for analysis

R-WP3/D503-71 The safety tool shall allow to record hypothesis used in the safety analysis

R-WP3/D503-72 The safety tool shall allow to record known limits of the safety analysis

R-WP3/D503-73 The safety tool shall allow to record the level of confidence on used data for the safety analysis

R-WP3/D503-74 The traceability between all the safety requirements and the hazard shall be verified.

R-WP3/D503-75 The safety tool must be able to ensure that all identified hazards have adequate and proven mitigation coverage.

R-WP3/D503-76 The traceability between safety requirements and the test case shall be established

The next sub section explains the way data are exchanged when performing the Model Based Safety Analysis.

5.2 Data Exchange between the Requirements Management tool and the Safety tool

This section deals with the tools used during the Model Based Safety Analysis process and the way data are exchanged between tools. Model Based Safety Analysis also requires the Requirements management tool DOORS to manage requirements, the Requirements Interchange Format tool ReqIF to perform data exchange and the Safety Analysis tool dedicated to the safety analysis.

The data Exchanged between DOORS and Safety Architect are described in the Table 15.

PURPOSE	Show the way data are exchanged between the requirements management tool and the safety tool.
INPUTS	System requirements
OUTPUTS	Safety requirements
DESCRIPTION	The safety Engineer: • Imports the requirements from the DOORS format to the ReqIF format • Imports the requirements from the ReqIF format to the Safety Architect format • Performs Safety Analysis • Exports the safety requirements from the Safety Architect format to the ReqIF format • Exports the safety requirements from the ReqIF format to the DOORS format
TOOLS	DOORS, Safety Architect

Table 15 Data exchange scenario between DOORS and Safety Architect

The Figure 8 gives an illustration of the data exchanged between the requirements management tool and the Safety Analysis tool via the requirements interchange format ReqIF.

Figure 8 Data Exchange between the requirements management tool and the Safety Analysis tool via the requirements interchange format ReqIF

The following IOS design requirements are identified when exchanging data between DOORS and Safety Architect:

R-WP3/D503-77 The Safety Engineer shall be able to import existing safety requirements from the DOORS format to the ReqIF format

R-WP3/D503-78 The Safety Engineer shall be able to import safety requirements from the ReqIF format to the Safety Architect format

R-WP3/D503.030-79 The Safety Engineer shall be able to export the refined safety requirements from the Safety Architect format to the ReqIF format

R-WP3/D503.030-80 The Safety Engineer shall be able to export the refined safety requirements from the ReqIF format to the DOORS format

R-WP3/D503.030-81 The following requirements shall be fulfilled: **R-WP3/D503.030-29, R-WP3/D503.030-30**

R-WP3/D503.030-82 The transformation from the DOORS format to the ReqIF format must be hundred percent succeeded

R-WP3/D503.030-83 The transformation from the ReqIF format to the DOORS format must be hundred percent succeeded

R-WP3/D503.030-84 The transformation from the ReqIF format to the Safety Architect format shall be hundred percent succeeded

R-WP3/D503.030-85 The transformation from the Safety Architect format to the ReqIF format must be hundred percent succeeded

R-WP3/D503.030-86 The data exchange between the tools shall be secured

R-WP3/D503.030-87 The data exchange between ReqIF, DOORS and Safety Architect shall be quickly performed. This requirement depends on the data size that we have to import/export.

The next section defines the terms and the abbreviations used in this deliverable.

6 Terms, Abbreviations and Definitions

CRYSTAL	CRitical SYSTem Engineering AcceLeration
ASAP	Advanced System Architecture Program
ASM	Accident Scenario Model
DOORS	Dynamic Object Oriented Requirements System
FMEA	Failure Mode and Effects Analysis
HBS	Hazard Breakdown Structure
IF	Inefficiency Factors
IOS	Interoperability Specification
OC	Operational Context
PHA	Preliminary Hazard Analysis
RMT	Requirement Management Tool
RRF	Risk Reduction Factor
RTP	Reference Technology Platform
S	Severity
SAE	Safety Assurance Engineer
SAM	Safety Assurance Manager
SDM	System Dysfunctional Model
SHA	System Hazard Analysis
SHL	Safety Hazard Log
SIL	Safety Integrity Level
SSHA	Sub System Hazard Analysis
SysML	Systems Modelling Language
THR	Tolerable Hazard Rate
UML	Unified Modelling Language

Table 6-1: List of Acronyms

Accident	Uninspected even that results in the death or injury of personnel, system loss, or damage to property, equipment or the environment
Barrier	Actor that prevents a hazard (operational or technical) from developing into a Railway hazard and finally a potential accident. A barrier can be implemented by: • A procedure (e.g. operational and/or maintenance procedures with trainings of the involved staff). • A Function
Failure	Inability of a system, subsystem, or component to perform its required function
Failure cause	Process or mechanism responsible for initiating the failure mode. The possible processes that can cause component failure include physical failure, design defect, manufacturing defects, and environmental forces.
Failure effects	Consequence (s) a failure mode has on the operation, function, or status of an item and on the system.
Failure mode	The manner by which an item fails
Failure mode and Effects analysis	Tool for evaluating the effects of potential failure modes of subsystem, assemblies, components, or functions. It is primary the reliability tool to identify failure mode that would adversely affect overall system reliability. FMEA has the capability to include failure rates for each failure mode in order to achieve a quantitative probabilistic analysis.
Fault	Undesired anomaly in the functional operation of an equipment or system.
Fault tree	Model that logically and graphically the various combinations of possible events occurring in a system that leads to a previously identified hazard or undesired event
Fault tree analysis	System analysis technique used to determine the root cause and the probability of occurrence of a specified undesired event.
Function	A mode of action or activity by which a product fulfils its purpose.
Hazard	Any real or potential condition that can cause injury, illness, or death to personnel; damage to or loss of a system, equipment or property; or damage to the environment.
Hazard Breakdown structure	Lists all potential accidents to be considered, the operational contexts in which they may occur, and links them (through a consequence analysis) with the Railways and technical hazards that could lead to such potential accidents.
Mitigation	Action taken to reduce the risk presented by a hazard, by modifying the hazard in order to decrease the incident probability and or the incident severity. Mitigation is generally accomplished through design measures, use of safety devices, warning devices, training or procedures. It is also referred to as hazard mitigation and risk mitigation.
Operational context	Feature where, when and how a Railway Hazard may develop into a potential Accident
RRF	Risk Reduction factor is a factor that allows reducing the probability of an accident occurring. It takes into account a specific operational context or the presence of a protection function (barrier).
Safety barrier	A system or action, intended to reduce the rate of a Hazard or a likely Accident arising from the Hazard and/or mitigate the severity of the likely Accident.

Safety requirements	Safety characteristics (qualitative or quantitative) of a system and its operation (including operational rules) necessary to meet legal or company safety targets
SIL	A number which indicates the required degree of confidence that a system will meet its specified safety functions with respect to systematic failures.

7 References

[HGomaa]	Hassan Gomaa ; SOFTWARE MODELING AND DESIGN - UML, Use Cases, Patterns, and Software Architectures, 2011
[ECHATSS]	Ericson Clifton A, Hazard Analysis Techniques for System Safety, 2005
[KWSOREQ]	Karl E Wiegers, Joy Beatty ,Software Requirements, 2013

8 Annex

D503.010 - Use case definition

2013-11-15_CRYSTAL_UC_Definition_WP5

D503.020 - IOS needs for RTP specification

D503.020 - IOS needs for RTP specifi

D604.011 - Specification, Development and Assessment for Safety Engineering

CRYSTAL_D_604_011_v0-05_RB-2.docx