



Deliverable D2.1-Modelling guidelines and Moving Block Use Cases characterization

C. Seceleanu, F. Flammini, S. Marrone, F. Mogavero, R. Nardone, L. Starace, V. Vittorini, Rim Saddem-Yagoubi, Mohamed Ghazel, Julie Beugin, et al.

► To cite this version:

C. Seceleanu, F. Flammini, S. Marrone, F. Mogavero, R. Nardone, et al.. Deliverable D2.1-Modelling guidelines and Moving Block Use Cases characterization. Mälardalen University. 2021. <hal-04487984>

HAL Id: hal-04487984

<https://hal.science/hal-04487984v1>

Submitted on 5 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



Deliverable D 2.1

Modelling Guidelines and Moving Block Use Cases Characterization

Project acronym:	PERFORMINGRAIL
Starting date:	01/12/2020
Duration (in months):	31
Call (part) identifier:	S2R-OC-IP2-01-2020
Grant agreement no:	101015416
Due date of deliverable:	Month 7
Actual submission date:	29-06-2021
Responsible/Author:	MDH/Cristina Seceleanu
Dissemination level:	PU
Status:	Issued

Reviewed: (no)

Document history		
Revision	Date	Description
0.1	25-03-2021	First issue.
0.2	30-05-2021	Stable method of selection of OSs. Defined content of Sections 6, 7, and 8.
0.3	18-06-2021	First version for internal review.
1.0	28-06-2021	Final version for JU review.

Report contributors		
Name	Beneficiary Short Name	Details of contribution
Cristina Seceleanu Francesco Flammini	MDH	<ul style="list-style-type: none"> • Table of contents • Executive Summary • Introduction (Section 1) • Contribution to Behavioural Modelling (Section 5.3) • Conclusions (Section 8) • OS#7, OS#9 • Review of Sections 5.2, 5.3
Stefano Marrone Fabio Mogavero Roberto Nardone Luigi Libero Lucio Starace Valeria Vittorini	CINI	<ul style="list-style-type: none"> • Relevant Scenarios (Section 3) • Structural Modelling (Section 5.2) • Contribution to Behavioural Modelling (Section 5.3) • Recommendations for integration in CENELEC process (Section 7) • OS#4, OS#5 • Review of Sections 3, 4
Julie Beugin Mohamed Ghazel Rim Saddem	UNI EIFFEL	<ul style="list-style-type: none"> • Methodology for System Modelling and Analysis (Section 4) • Requirement modelling (Section 5.1) • Contribution to Structural Modelling (Section 5.2) • Contribution to Behavioural Modelling (Section 5.3) • Hazard Modelling (Section 5.4) • OS#2, OS#6 • Review of Sections 5.1
Rob M.P. Goverde Nina Versluis	TUD	<ul style="list-style-type: none"> • Initial Scenario Set (Section 3.1.2) • OS#3, OS#10 • Review of Sections 1, 2

Bob Janssen	EULYNX	<ul style="list-style-type: none"> Guidelines for Data Modelling (Section 6) OS#1 Review of Sections 6, 7, 8
Miquel Garcia	ROK	<ul style="list-style-type: none"> Contribution to Hazard Modelling (Section 5.4)
Mohamed Samra Achila Mazini	UoB	<ul style="list-style-type: none"> Background (Section 2) OS#8

Internal Reviewers	
Name	Beneficiary Short Name
Massimo Benerecetti	CINI
Egidio Quaglietta	TUD

Funding

This project has received funding from the Shift2Rail Joint Undertaking (JU) under grant agreement No 101015416. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Shift2Rail JU members other than the Union.

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author's view – the Joint Undertaking is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

Table of Contents

Executive Summary	6
Abbreviations and Acronyms.....	7
1. Introduction	9
1.1 Objectives and Scope	9
1.2 Related Documents.....	9
2. Background	11
3. Relevant Scenarios.....	13
3.1 Selection of Relevant Scenarios	15
3.1.1 Method of Selection	15
3.1.2 Initial Scenario Set	17
3.1.3 Survey-based analysis of industrial relevance.....	18
3.1.4 Complexity and feasibility analysis.....	20
3.2 Description of Operational Scenarios	21
3.2.1 Short Description of Operational Scenarios	21
3.2.2 Template for operational scenario detailed description.....	25
4. Methodology for System Modelling and Analysis.....	30
4.1 Workflow.....	30
4.1.1 MB Requirements Engineering.....	30
4.1.2 MB Structural Modelling.....	32
4.1.3 MB Behavioural Modelling	32
4.1.4 Verification & Validation	34
4.1.5 Hazard Modelling.....	36
4.2 Modelling Principles and Artefacts	37
5. Guidelines for System Modelling.....	39
5.1 Requirements Modelling.....	39
5.2 Structural Modelling	41
5.3 Behavioural Modelling	53
5.3.1 Semi-formal Models	54
5.3.2 Formal Models	59
5.4 Hazard Modelling.....	70
6. Guidelines for Data Modelling.....	74
6.1 Modelling attributes and relations	74
7. Recommendations for Integration in CENELEC Process.....	78
8. Conclusions	80

References	81
Appendix A – OS#1 - Trackside initialisation	86
Appendix B – OS#2 - Start of Mission	90
Appendix C – OS#3 - Points Control	100
Appendix D – OS#4 - Crossing of Radio Hole	105
Appendix E – OS#5 - Loss/Restore of Communications	109
Appendix F – OS#6 - Loss of Train Integrity	114
Appendix G – OS#7 - Shunting Movement	120
Appendix H – OS#8 - End of Mission	124
Appendix I – OS#9 - Supervising Distance in Normal VCTS Driving	129
Appendix J – OS#10 - Splitting of a VCTS Initiated by Slave	132
Appendix K – Survey questions	135

Executive Summary

The present document constitutes the Deliverable D2.1 “Modelling guidelines and Moving Block (MB) Use Cases characterization”, which is part of Work Package 2 of the “PERformance-based Formal modelling and Optimal tRAffic Management for movING-block RAILway signaling” project (acronym: PERFORMINGRAIL).

This deliverable describes the selection method and gives the details of relevant operational scenarios of moving block (MB) systems, including virtual coupling (VC) configurations, in terms of parameters, possible variants to be considered in the development of models, as well as potential hazards. The information has been extracted from the public documents available and from the feedback received from experts to the extent that this was possible in the current state of our knowledge. Therefore, the definition of the operational scenarios presented in the deliverable constitutes a starting point that can be updated and improved in future tasks of WP2.

In addition, the deliverable identifies and exemplifies on toy railway-relevant examples the appropriate modelling and verification notations and formalisms that are able to capture the structural and behavioural aspects of MB and VC scenarios and configurations, as well as those applicable to the hazard analysis of the latter. The work of this deliverable builds on the results of previous S2R work carried out in ASTRAIL, MOVINGRAIL, as well as S2R X2Rail-1 and S2R X2Rail-3, as mentioned in Section 1.2.

This deliverable makes the following contributions:

- Identification of operational scenarios of moving block systems, including virtual coupling, for various use cases.
- A template for the systematic textual description of the scenarios, in which the applicable use cases, performance indicators, functional components, parameters, variants, behaviour and hazards can be textually specified.
- A generic methodology for building structural and behavioural models for MB and VC systems, and verifying them.
- A description of the selection method of the most critical and industrially relevant operational scenarios, the analysis of their industrial relevance based on a survey used to collect feedbacks from stakeholders, and their complexity and feasibility analysis.
- Overviews of languages, notations and frameworks deemed appropriate for the structural modelling, as well as semi-formal and formal modelling of MB and VC systems behaviour and hazards.
- Guidelines and naming conventions for data modelling.
- Instantiations of the defined operational scenario template for 10 relevant MB and VC scenarios (see the Appendixes).

Abbreviations and Acronyms

Abbreviation / Acronyms	Description
ATP	Automatic Train Protection
BDD	Block Definition Diagram
CBTC	Communication Based Train Control
CCA	Cause Consequences Analysis
CPN	Coloured Petri Net
CRE	Confirmed Rear End
CSRE	Confirmed Safe Rear End
CTL	Computation Tree Logic
EoA	End of Authority
EoM	End of Mission
ERTMS	European Railway Traffic Management System
ETA	Event Tree Analysis
ETCS	European Train Control System
EVC	European Vital Computer
FMB	Full Moving Block
FMEA/FMECA	Failure Mode Effect Criticality Analysis
FTA	Fault Tree Analysis
GNSS	Global Navigation Satellite System
GSPN	Generalized Stochastic Petri Net
HAZOP	Hazard and Operability Analysis
IBD	Internal Block Diagram
IM	Infrastructure Manager
LS	Limited Supervision
LTL	Linear Temporal Logic
M2M	Model to Model
M2T	Model to Text
MA	Movement Authority
MB	Moving Block
MBSE	Model-Based System Engineering
NL	Natural Language
OMG	Object Management Group
OS	Operational Scenario
PHA	Process/Preliminary Hazard Analysis
PN	Petri Net
PR	Position Report
PTCTL	Probabilistic Timed Computation Tree Logic
RBC	Radio Block Centre
RE	Requirement Engineering
S2ML	System Structure Modeling Language
S2R	Shift2Rail
SAN	Stochastic Activity Network
SB	Standby Mode
SFE	Safe Front End

SIL	Safety Integrity Level
SM	State Machine
SMHA	State Machine Hazard Analysis
SoM	Start of Mission
SPTA	Stochastic Priced Timed Automata
SR	Staff Responsible
SRE	Safe Rear End
STPA	System Theoretic Process Analysis
TA	Timed Automata
TCTL	Timed Computation Tree Logic
TIMS	Train Integrity Monitoring System
TTD	Trackside Train Detection
UC	Use Case
UML	Unified Modeling Language
V&V	Verification & Validation
VB	Virtual Block
VC	Virtual Coupling
VCTS	Virtually Coupled Train Set
WP	Work Package

1. Introduction

1.1 Objectives and Scope

The main objective of this deliverable is to define and describe a generic methodology for modelling and analysing moving block systems, including virtual coupling, as well as guidelines for the development of parameterizable semi-formal and formal models amenable to formal analysis of the operation under moving block and virtual coupling configurations.

Concretely, in this deliverable we first identify a set of relevant operational scenarios for moving block and virtual coupling systems (Sections 3.1.1 and 3.1.2), after which we rank them (Section 3.1.3 and 3.1.4). Next, we introduce a template-based description of the scenarios, in Section 3.2, in which system variants, parameters and hazards are textually described. The template-based description of the relevant operational scenarios is reported in the Appendixes. This work helps the consortium to identify the features that need to be captured by the semi-formal and formal models, which will be developed in deliverable D2.2 of WP2. The work on operational scenario characterization and description will also facilitate the definition of the guidelines for system modelling with respect to requirements, structure, behaviour, and hazards, all described in Section 5, as well as with respect to data modelling, as presented in Section 6. The deliverable ends with a description of recommendation for integration into the CENELEC process in Section 7.

The scope of this deliverable is to set the premises and ease the work of developing generic semi-formal and formal models of moving block, instantiated to concrete system configurations including virtual coupling ones, which will be described in deliverable D2.2.

1.2 Related Documents

The work of this deliverable is based on the results from the work performed in previous S2R projects, in particular in ASTRAIL (“D2.1 - Modelling of the moving block signalling system” and “D2.2 - Moving Block signalling system Hazard Analysis”), MOVINGRAIL (“D1.1 Report on Moving Block Operational and Engineering Rules” and “D4.1 Market potential and Operational Scenarios of Virtual Coupling”), as well as on the work performed in S2R X2Rail-1 and S2R X2Rail-3, as reviewed in WP1 Task 1.1.

Relationship with other deliverables. This paragraph shows the relationships between D2.1 and other PERFORMINGRAIL deliverables. Figure 1 represents the dependency relationships between these documents.

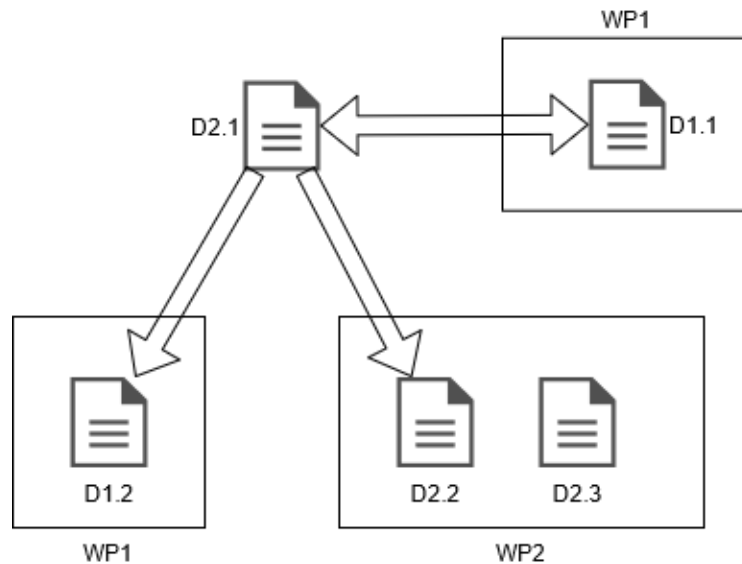


Figure 1. Relationships with other PERFORMINGRAIL deliverables

D2.1 is an input for the development of other documents. Specifically, the content of this deliverable will be used in:

- D1.2 (“Best practice, recommendations and standardisation to definition of the Railway Minimum Operational Performance Standards”) where the baseline to summarize the recommendations of all the WPs will be used;
- D2.2 (“Moving Block Specification Development”) where semi-formal and formal models on the basis of the modelling guidelines defined here will be developed;
- D2.3 (“Moving Block Verification and Validation”) where the developed semi-formal and formal models will be validated, also by using the operational scenarios defined in this document.

Furthermore, there is an intersection between the contents of this deliverable and of D1.1 (“Baseline system specification and definition for Moving Block Systems”) since the related tasks have been conducted in parallel. The connection points between the two deliverables are:

- Use case definition: the definition of the use cases that is present in D1.1 is used in this deliverable to build the Operational Scenarios that, according to the methodology here described, can be seen as specialisation of one or more use cases;
- Operational Scenario vs Use Case mapping: since one of the objectives of D1.1 is to detail the ETCS MB/VC behaviour within at least four use cases, the choice of which specific use case to model has been driven also by the necessity to maximise the coverage between OSs and UCs. This mapping has been added to D1.1 on the basis of information present in D2.1;
- Modelling Guidelines: the requirements, the architectural and the behavioural diagrams detailing the structure and the dynamics of ETCS MB/VC are reported in D1.1 in conformity to the SysML specification language. In the present deliverable, such a language has been assessed as one of the most suitable formalism to capture the high-level behaviour of a system.

2. Background

In modern railway control and signalling systems, it is very challenging to predict accurate system behaviour, interoperability, safety, and reliability. According to [D5.1_X2Rail-2], this is usually caused by the imprecise definition of system requirements that consequently complicates the system testing and verification. Also, vague requirements can lead to poor design choices and imprecise interfaces definition. Formal and semi-formal methods can be defined as a set of rigid mathematical practices to define software-intensive systems [D4.1_ASTRAIL]. Formal methods can be used in three application levels of the system development process, including system specification definition, development, and verification. The adoption of formal methods in railway signalling systems will help to fulfil several requirements on safety, security, and reliability, while analysing relationships between components, and verify system behaviour against the set of requirements identified.

[D5.1_X2Rail-2] listed some representative examples that demonstrate the benefit of using formal methods in railway signalling, in categories of formal verification, formal development and special-purpose applications. In terms of formal verification, railway infrastructure managers such as RATP, New York City Transit, and Stockholm Metro, has used formal methods to verify the safety of relay-based and computerised interlocking systems. Furthermore, several metro projects in Asia and Europe have used formal verification for safety assessment of the revenue service software for CBTC systems [D5.1_X2Rail-2]. The system-level safety properties of the CBTC system for the New York City Transits Line were also verified using the formal verification tools [D5.1_X2Rail-2].

As for the formal development, formal methods have been used to define safety-critical parts of software for CBTC systems. Representative projects of this example include Line 14 of the Paris subway, the automatic driverless shuttle of Roissy Airport, and Alstom's Urbalis 400 CBTC system [D5.1_X2Rail-2]. Formal methods have also been used to define train control systems and Automatic Train Protection (ATP) in several projects. As for special-purpose applications, a formal app called Ovado was used by Thales to validate configuration data for the zone controllers and train controllers of the CBTC system for RATP's Line 13. In addition, Bombardier used a formal app to verify the absence of certain run-time errors in their computerised interlocking systems [D5.1_X2Rail-2].

[D2.1_ASTRAIL] is one of the few pieces of literature available that modelled the moving block (MB) signalling system (without trackside train detection) using a semi-formal method, namely Unified Modelling Language (UML) state machine diagrams. The research focused on understanding the moving block system architecture, boundaries, interaction of system components and use cases to visualise the system behaviour. The use cases considered traffic type and density, environmental conditions, and Grade of Automation to cover a set of various scenarios including normal operations, degraded mode, transition phases, system initialization, and recovery from critical failure. [D2.1_ASTRAIL] is an essential starting point on the way to develop a set of formal and semi-formal specifications of the abstract moving block system behaviour.

The list of projects previously mentioned reflects the growing interest in the formal and semi-

formal methods and their contribution to the rail industry. However, there is no dominant mature technology that is considered suitable for all the stages of the signalling system development [D4.1_ASTRAIL]. [D4.1_ASTRAIL] analysed information from scientific literature, relevant projects, and railway practitioner to produce a ranking matrix that evaluates formal and semi-formal tools used to define rail systems in various development phases. The analysis showed dominance for the UML for the high-level representation of system models. The survey with the industry practitioners indicated a large variety of formal tools with preferences given to tools that support formal verification as well as system modelling.

On the other hand, the 4SECU Rail project has focused on performing a cost/benefit analysis for the adoption of formal methods in the railway environment. [D2.1_4SECU Rail] focused on prototyping a formal and semi-formal method Demonstrator, while defining its structure from methodologies and tools perspectives. [D2.1_4SECU Rail] completed the specification process with an application of the Demonstrator process to a signalling system case study.

This PERFORMINGRAIL deliverable will analyse the outcomes of the previously mentioned projects to provide guidelines for moving block and virtual coupling systems modelling to guide the choice in terms of notation and tools to be deployed in future tasks. To achieve that vision, this document will first present a structured, repeatable, and automatable process (i.e. template) that adopts the main software engineering principles. The template will be used to specify the different system configurations, by means of an automatable instantiation operation. Furthermore, this deliverable will characterise the moving block system, by defining a set of relevant operational scenarios highlighting the system parameters and possible variants to be considered in the development of the behavioural models.

3. Relevant Scenarios

Before any other description, the clear definitions of both use cases and operational scenarios are due. The concept of ETCS L3 use case (use case, for brevity) can be linked to the notion of procedure as described in the UNISIG's subset 026 describing the ERTMS L1/L2 [UNISIG_026]. In that document, procedures are defined as “the required reaction of the ERTMS/ETCS entities (subsystems and components) to either information exchanged between ERTMS/ETCS entities or events (triggered by external entities or internal events). The procedures focus on the required change in status and mode of the described ERTMS/ETCS entities”. The main difference between UNISIG procedures and use cases, as they are used in this document, is that use cases do not necessarily deal with state/level transitions, as they are wider in their scope.

Starting from this definition, let us define the concept of operational scenario. An operational scenario is a concrete sequence of actions/events of the ETCS entities and external actors in a specific railway configuration and with the objective to evaluate some indices (e.g., performance, availability, safety). In practice, an operational scenario can be defined across multiple use cases. The definition of operational scenarios is inspired by [ISO_29148], where they are defined as “an imagined sequence of events that includes the interaction of the product or service with its environment and users, as well as interaction among its product or service components”. Another related literature source for scenario concept definition is [EEIG_ERTMS], where they are defined as sequences of steps of ERTMS/ETCS components' interactions. Figure 2 depicts these concepts and their mutual relationships.

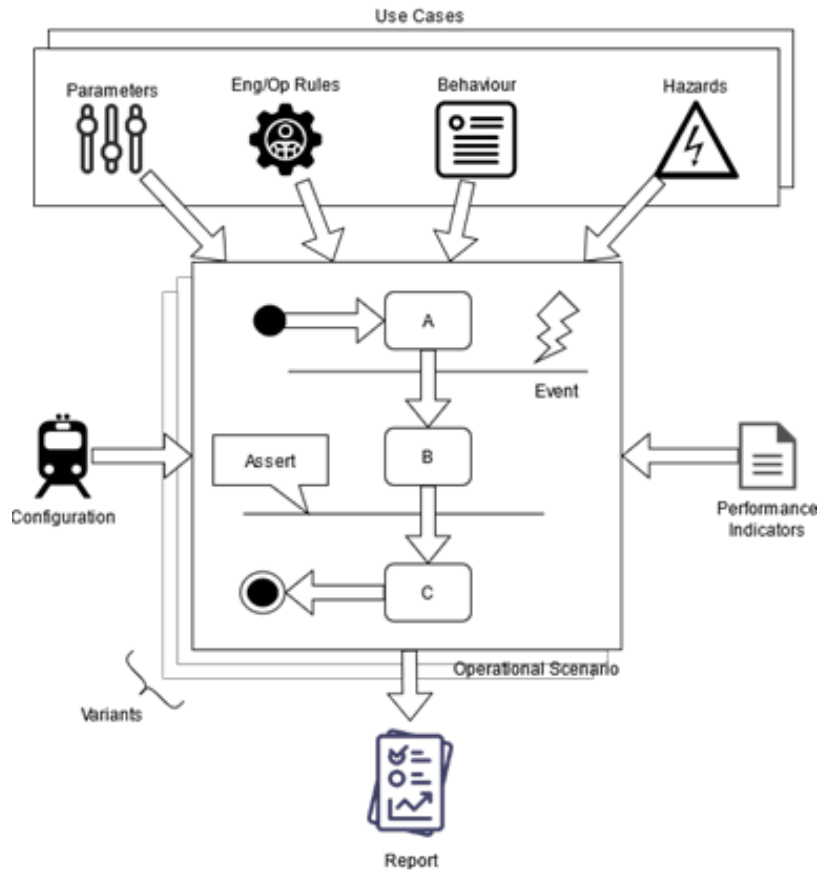


Figure 2. Operational Scenario concept and relations.

More formally, an operational scenario takes the following information as inputs:

- Use case: the description of the use cases applicable to the operational scenario. Each use case is described by four main elements.
 - Behaviour: the description of the actions and reactions of the ETCS L3 systems (both on-board and trackside) as well as the related external subsystems and actors (e.g., Driver, Dispatcher, Traffic Management System, etc.);
 - Parameters: the specific variable affecting functional and non-functional aspects of the use case (e.g., length of the train, duration of the mute timer);
 - Engineering/Operational rules: the set of best practices, the rules behind the determination of the values of some parameters and the manual procedures used to recover the ETCS L3 system in case of (partial) failures;
 - Hazards: dangerous situations that require investigation of the conditions under which they occur during the operational scenario.
- Configuration: as the operational scenario must be concrete to be evaluated, the mere concatenation of actions related to the use cases is not sufficient since the results of such an evaluation can change significantly, depending upon different conditions. These conditions may include:
 - presence/absence of Trackside Train Detection systems,
 - diverse signalling systems and market segments (heavy rail, metro systems, high speed/capacity lines);

- schema of the railway track used for the operational scenario (shape and length of the track, positions of points, switches, etc.).
- Variants: alternative versions of the scenario may be included, where small changes are possible that do not affect the main behaviour of the scenario itself. An example of such scenario is to consider the exit of the train from the Radio Hole Area determined by a PR with a (i) MSFE (ii) CSRE out of the area. In this context, a nominal case and at least an alternate variant must be reported.
- Performance indicators are observable quantities that allow for evaluating whether operational scenarios reach the intended threshold; requirements are needed to understand the goals of such an evaluation in terms of which are the variables to observe, and which are the targets to reach for such variables. Performance indicators could be qualitative or quantitative. They could belong to the following categories:
 - Logical: assertion made on system variables that the system must fulfil (e.g., invariants, pre/post-conditions). The map to pass-/fail-criteria in system tests.
 - Functional: system property directly derived from functional system requirements.
 - Availability/Reliability: indices related to availability/reliability.
 - Safety: evaluation of hazard conditions and/or verification that protection mechanisms guarantee a safe running of the trains.
 - Performance: indices related to performance in strict terms (e.g., line capacity, mission time, headways, etc.).

An operational scenario is a sequence of **steps** (i.e., A, B, C, in Figure 2); the transition from one step to another can be determined by external as well as internal **events** (i.e., the pushing of a button by the Driver or the reception of a certain message by ETCS L3 Trackside). Furthermore, an operational scenario can be enriched with **assertions** checking that a certain property is globally or locally held by the entities participating in the scenario. As described in the following, a formalisation of operational scenarios is out of the scope of this deliverable while Sections 5 and 6 report a review of the most suitable formalisms for modelling the ERTMS MB system.

The output of the performance analysis is summarized in a report describing (a) if a required performance indicator is matched or not, and/or (b) what the optimal set of parameter values in a certain configuration is, on the basis of a sensitivity analysis carried out on one or more quantitative performance indicators.

3.1 Selection of Relevant Scenarios

3.1.1 Method of Selection

This subsection is devoted to the description of the methodology behind the definition and the description of the operational scenarios as well as their relationship with the work of the other tasks in PERFORMINGRAIL. Figure 3 summarises the schema of the overall approach as it considers other PERFORMINGRAIL tasks.

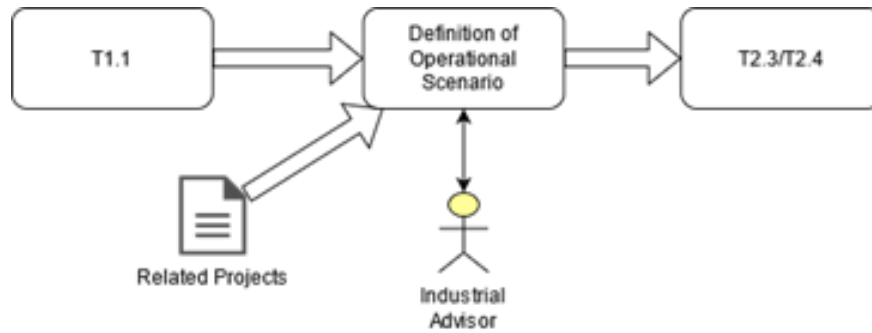


Figure 3. Operational scenario and relations with other tasks.

The Definition of the Operational Scenarios takes as input the definition of the use case as they are defined in Task 1.1 as well as applicable documents and related S2R projects deliverables. Some of these applicable documents are: MOVINGRAIL's "D1.1 Report on Moving Block Operational and Engineering Rules" and "D4.1 Market potential and Operational Scenarios or Virtual Coupling"; ASTRAIL's "D2.2 - Moving Block signalling system Hazard Analysis" and "D2.1 - Modelling of the moving block signalling system"; the work of 4SECRail.

The output will be a set of operational scenarios that will be modelled and analysed across the other tasks of the WP2: furthermore, the definition of a meaningful set of scenarios will also drive the work of other WPs. This set of scenarios should meet the following requirements:

- according to the Grant Agreement, the modelling and analysis have to be performed on at least 4 operational scenarios, allowing to delineate the moving block behaviour and analyse at least 3 different system configurations on the selected scenarios (i.e., diverse signalling systems on diverse market segments) against potential hazards;
- the scenarios should be feasible to model and to analyse, according to the modelling methodologies defines in T2.1;
- the scenarios should be relevant from an industrial point of view, as they should address real world needs.

Figure 4 details the activities of this task showing the methodology adopted to define the operational scenarios.

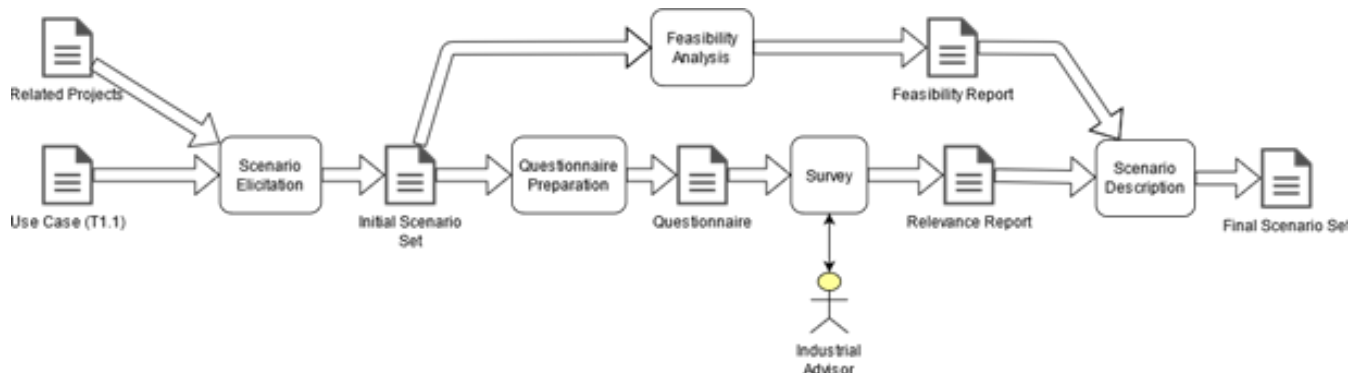


Figure 4. Operational Scenario Definition - task structure.

According to this schema, the following sub-activities are defined.

- Scenario elicitation: a set of brainstorming meetings to extract an Initial Scenario Set (10 scenarios).
- Questionnaire Preparation: definition of a questionnaire to submit to industrial partners (Advisory Board members and to WP Leaders of both X2RAIL-3 and X2RAIL-5 projects) in order to obtain a proper feedback on the initial work.
- Survey: the designed questionnaire is submitted to the industrial partners, who are asked to assign a relevance ranking to each of the scenarios in the Initial Scenario Set. The relevance is an index expressing the degree of interest of the industrial community in the proposed scenarios.
- Feasibility Analysis: an internal analysis conducted to verify the feasibility of the scenarios under the modelling and the analysis aspects.
- Scenario Description: the Initial Scenario Set is refined, by possibly modifying existing scenarios and ordering them according to the Feasibility Report and Relevance Report. A Final Scenario Set is produced as a final output of the task detailing the scenarios according to the schema presented above.

In addition to the requirements set in the Grant Agreement for the modelling and the verification of the operational scenarios, further guidelines will be followed.

- When variants of operational scenarios are included, the modelling activity can be performed on the nominal case as well as on another alternate case without necessarily considering all the versions.
- The evaluation of the impact made by industrial partners will not be mandatory for the choice of the scenarios that will be modelled and analysed. The actual choice will be made on the basis of both feasibility analysis and industrial relevance, and with the goal of covering the largest possible portion of the system specification.

3.1.2 Initial Scenario Set

Operational scenarios are based on use cases. The deliverable D1.1 (“Baseline system specification and definition for Moving Block Systems”) described a large number of use cases for moving block and virtual coupling as considered in other European projects, specifically X2RAIL-3, ASTRail, MOVINGRAIL, and EUG. In particular, 22 use cases for moving block were identified and 13 use cases for virtual coupling. From these sets, 10 use cases should be selected to be worked out into operational scenarios. This initial set of operational scenarios are then the basis for the final selection of operational scenarios that will be used for modelling in later tasks within WP2.

The main criterion for the selection of the initial operational scenario set was industrial relevance. It was expected that the final operational scenarios would include () one virtual coupling scenario. Therefore, it was decided that the initial set would be based on 8 moving block use cases and 2 virtual coupling use cases.

The initial selection of the moving block use cases was then based on the following criteria

- Industrial relevance
- Basic to advanced nontrivial scenarios
- Complexity in terms of number of Requirements, Operational Rules and Engineering Rules
- Presence of safety hazards
- Issues identified in MOVINGRAIL D1.1
- Include GNSS issues

Five moving block use cases were covered by several other use cases as described in D1.1, so these were not considered. An example of such use case is the On-Sight Movement that relates in ETCS L3 to the Sweeping functionality (D1.1).

For Virtual Coupling System Requirements and Operational and Engineering Rules were not available. Therefore, in this case it was decided to select one basic and one advanced nontrivial scenario for virtual coupling.

Based on these criteria the following initial set of use cases to be considered in defining the operational scenarios was obtained:

Moving block use cases for operational scenarios

1. Trackside Initialisation
2. Start of Mission
3. Normal Train Movement
4. End of Mission
5. Shunting
6. Loss/Restore of Communications
7. Loss of Train Integrity
8. Points Control

Virtual coupling use cases for operational scenarios

9. Supervising Distance in normal driving
10. Splitting of a VCTS - Initiated by Slave

This initial list was discussed and approved in a meeting with all PERFORMINGRAIL WP2 partners.

From this set three operational scenarios (or variants) may include GNSS issues: Normal Train Movement, Loss/Restore of Communications, and Loss of Train Integrity.

To build the list of potential operational scenarios, each single use case has been considered as a reference. One of the operational scenarios involves both the use cases Normal Train Movement and the Radio Hole, that is not in the list of the selected use case. Hence, the final list of operational scenarios is the following:

- OS#1 - Trackside Initialisation
- OS#2 - Start of Mission

- OS#3 - Points Control
- OS#4 - Crossing of a Radio Hole
- OS#5 - Loss/Restore of Communications
- OS#6 - Loss of Train Integrity
- OS#7 - Shunting Movement
- OS#8 - End of Mission
- OS#9 - Supervising Distance in Normal VCTS Driving
- OS#10 - Splitting of a VCTS Initiated by Slave

For each scenario, we preliminarily defined the abstract, a short description and a set of performance indicators, as described in the following of this document.

3.1.3 Survey-based analysis of industrial relevance

To evaluate the industrial relevance of the operational scenarios, an assessment survey has been set and submitted to Advisory Board members and to WP Leaders of both X2RAIL-3 and X2RAIL-5 projects.

For each operational scenario, the survey reported the abstract, a short description, and the set of performance indicators (i.e., the final objective of the evaluation). For each scenario, responders were asked to provide their evaluation (choosing among very high, high, medium, low, very low) according to the following criteria:

- Significance for market segments/signalling systems
- Safety challenges
- Industrial relevance

Moreover, open-ended questions are included for each operational scenario, allowing responders to add any additional comment/feedback/suggestion (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters). The survey ends by asking each responder to suggest the “best-4” operational scenarios, corresponding to the 4 most meaningful scenarios to be modelled and verify by means of formal approaches.

The survey agreed upon among all PERFORMINGRAIL WP2 partners, was sent to the industrial partners on the 27th of May, 2021, and closed on 10th June, 2021. The questionnaire is reported in Appendix K.

We received 6 anonymous answers that are reported in the following table. The table is ordered according to the final ranking. Starting from the left to the right columns, the table reports:

- Scenario ID;
- Scenario name;
- Significance for the moving block signalling system: for each sub-column, number of received preferences from very low (VL) to very high (VH);

- Impact on the system safety: for each sub-column, number of received preferences from very low (VL) to very high (VH);
- Industrial relevance of the evaluation: for each sub-column, number of received preferences from very low (VL) to very high (VH);
- Number of times the scenario has been selected as meaningful by responders.

OS id	OS title	significance for the MB signaling system					impact on the system safety					industrial relevance of the evaluation					selected as meaningful
		VL	L	M	H	VH	VL	L	M	H	VH	VL	L	M	H	VH	
OS#6	Loss of Train Integrity					6					6			1		5	6
OS#3	Points Control				1	5				1	5			1		5	5
OS#5	Loss/Restore of Communications				3	3				1	5			1	3	2	4
OS#1	Trackside Initialization					6				1	5				3	3	4
OS#2	Start of Mission				2	4				1	5			1	3	2	3
OS#7	Shunting Movement		1	3	1	1			4	1	1		2	1	1	2	1
OS#9	Supervision Distance in Normal VCTS Driving	1		2	2	1			1	1	4		2	2	1	1	1
OS#8	End of Mission			2	2	2			2	4			2	1	1	2	0
OS#10	Splitting of a VCTS Initiated by a Slave	1		4		1			2	1	3		2	3		1	0
OS#4	Crossing of a Radio Hole		2	2	2			2	1	1	2		2	2	2		0

The operational scenario Loss of Train Integrity has been selected by all the interviewed people as one of the most meaningful scenarios. In fact, all responders agree on its very high significance for the moving block signalling system and its very high impact on the system safety. Moreover, 5 out of the 6 consider this scenario as having a very high industrial relevance. Hence, it ended up the first in the final ranking.

The second scenario in the ranking is Points Control, which has been selected by 5 people. The same people consider this scenario as having a very high significance for the moving block signalling system, a very high impact on the system safety, and a very high industrial relevance.

In the third place, the two operational scenarios Loss/Restore of Communications and Trackside Initialization received 4 preferences. Both of them are considered as having a very high impact on the system safety, even if they are not considered as relevant for the industry as the previous ones.

Scrolling down the ranking, the operational scenario Start of Mission received 3 preferences as one of the “best-4” scenarios. Even though 5 people agree on its very high impact on the system safety and 4 agree on the very high significance for the moving block signalling system, also this scenario is not considered as relevant for the industry either.

Then, the two scenarios Shunting Movement and Supervising Distance in Normal VTCS Driving received 1 preference. Both of them are considered of average significance for the system. Moreover, the former has a higher industrial relevance, while the latter has a higher impact on the system safety.

At last, the three scenarios End of Mission, Splitting of a VCTS Initiated by Slave, and Crossing of a Radio Hole were not selected by anyone as one of the “best-4” scenarios for formal modelling and verification.

3.1.4 Complexity and feasibility analysis

To analyse the complexity and to obtain a feasibility ranking of the operational scenario, the following table reports, for each scenario, the number of applicable Use Cases, the number of components and the number of functions involved.

These three parameters could represent a preliminary analysis of the complexity of each scenario, that would be proportional to the effort needed in the future formal modelling and analysis phases. At last, the last column shows the types of Performance Indicators (i.e., quantitative and/or qualitative). Of course, this analysis cannot give the final complexity evaluation, which can be refined only in the formal modelling and analysis phases of selected scenarios (objective of the later tasks within WP2).

OS id	OS title	# Use Cases	# Components	# Functions	PI types
OS#3	Points Control	3	5	9	quantitative
OS#10	Splitting of a VCTS Initiated by a Slave	3	4	7	quantitative
OS#6	Loss of Train Integrity	4	6	6	quantitative
OS#2	Start of Mission	3	5	6	quantitative
OS#8	End of Mission	1	5	6	quantitative
OS#1	Trackside Initialization	2	7	5	quantitative
OS#9	Supervision Distance in Normal VCTS Driving	1	4	5	quantitative
OS#4	Crossing of a Radio Hole	4	6	4	quantitative
OS#5	Loss/Restore of Communications	5	6	4	quant. / qualit.
OS#7	Shunting Movement	4	6	4	quantitative

The table is ordered by column *#Functions* that represents, in this preliminary phase, an indicator of the complexity of the behavioural modelling of such scenarios and it could be considered as directly proportional to the computational complexity of the solution of the formal models.

According to this ranking, the scenario Points Control involves the highest number of functions. In the second place, the scenario Splitting of a VCTS Initiated by a Slave involves 7 functions.

Scrolling down the ranking, the scenarios Loss of Train Integrity, Start of Mission and End of Mission involve 6 functions. Among them, Loss of Train Integrity has 4 applicable Use Cases and involve 6 components.

The two scenarios Trackside Initialization and Supervising Distance in Normal VCTS Driving involve 5 functions. At last, the scenarios Crossing of a Radio Hole, Loss/Restore of Communications and Shunting Movement involve 4 scenarios.

3.2 Description of Operational Scenarios

3.2.1 Short Description of Operational Scenarios

The 10 Operational Scenarios before introduced, are here described. For each scenario, a short abstract, a description and the list of performance indicators are given.

- OS#1 - Trackside initialisation

Abstract:

This scenario describes the process of initialising the trackside control systems with up-to-date values.

Description:

The concept of state vector and its initialisation is central to this scenario. Trackside in this context is the area under control of an RBC or more general, of a Central Safety System that combines RBC and interlocking.

Trackside area is an area that can be located, both geographically and in terms of track topology.

State space represents the status of the trackside system. This is represented by a state vector, i.e. a vector of (object, state)-tuples. Objects include fixed trackside elements such as points as well as transient objects such as trains and temporary speed restrictions.

State vector initialisation allocates state to the values. State is detected through sensors, actuators and messages.

Performance indicators:

- Average startup time - the time the system needs to reach operational status (availability, quantitative);
- Completeness - the probability that an object-status remains unknown (reliability & safety, quantitative);
- Safety - the probability that a vital object state value is detected wrongly (safety, quantitative).

- OS#2 - Start of Mission

Abstract:

This scenario concerns the Start of Mission (SoM) of a non-localised train followed by Staff Responsible to re-locate the train. When the driver begins the Start of Mission procedure, the train Position Report (PR) status is "Unknown" and L3 Trackside authorizes the train to run in Staff Responsible mode until the train reaches a first location reference and reports its position to the L3 Trackside.

Description:

A stopped train in Stand-by mode and under the supervision of the L3 system has to start a mission with a SoM procedure to reach Full Supervision, On-Sight, Limited Supervision, or Staff Responsible modes. When the SoM procedure is launched, the train cab desk is

assumed to be already open by the driver and, no communication session is still established or being established between the on-board and trackside parts.

The final aim of this operational scenario is to allow the analysis of the success/failure to obtain a first correct and valid position after the SoM procedure. This can be especially analysed when the SoM procedure is started somewhere on a line only equipped with virtual balises. Virtual Block (VB) hazards are related to GNSS-based VB reader issues, in particular, issues related to GNSS feared events.

Performance indicators:

- Probability for the first position to be erroneous while L3 trackside receives a valid PR (i.e. format correct but real position wrongly bounded) (safety, quantitative).

- OS#3 - Points Control

Abstract:

This scenario concerns the moving, locking and releasing of points related to two subsequent trains requesting to pass over different points.

Description:

In this scenario, the situation is considered in which two trains running under normal moving block conditions cross a point consecutively, with the second train requiring the point to move to a different position.

This point cannot be moved as long as the first train occupies the associated track area. Also, the point cannot be moved when the point is already reserved to the second train. These point movement timing restrictions apply due to the hazard of moving a point while a train is passing, or about to pass, over it, possibly leading to derailment of the train.

Performance indicators:

- Headway time - minimum time between train heads over points (performance, quantitative).

- OS#4 - Crossing of Radio Hole

Abstract:

A connected train moving under the supervision of ETCS L3 enters an active Radio Hole area, in which a blackout in communications is expected.

Description:

The final aim of this operational scenario is to evaluate the time needed by a train to cross a radio hole depending on parameters such as the speed of the train, the quality of the communication network and the radio hole timer. Two cases may occur: in the first one, the parameters are set to values that guarantee the trackside to keep the connection of the train alive during all the disconnection interval; in the second case, the connection with the train is lost and the train does not reconnect to the trackside in a timely manner: in this case a SR exit from the radio hole can be necessary.

Performance indicators:

- Radio Hole average crossing time - average time for the train to cross the radio hole (performance, quantitative).

- OS#5 - Loss/Restore of Communications

Abstract:

This scenario analyses the system behaviour in case of loss of communication against known and not known hazards and/or understanding the sweeping activation conditions.

Description:

In the case of a connected ETCS L3 supervised train, if communication with the train is lost, three possible cases can occur:

(A) connection is re-established before session timeout;

(B) connection is re-established before session timeout, with changes in train position/id/length;

(C) the train fails to re-connect before session timeout.

Performance indicators:

- Hazard Probability - probability of hazard in case of loss/restore of communication (safety, quantitative);
- Set of sweeping conditions - set of conditions bringing to sweeping procedure activation (performance, qualitative).

- OS#6 - Loss of Train Integrity

Abstract:

In this operational scenario, a connected train moving under the supervision of ETCS L3 train loses its integrity.

Description:

The final aim of this operational scenario is to protect the rear end of the train and other trains from collision in the case that a train has lost its integrity. This may occur for different reasons but in the event that a train splits unintentionally, the Dispatcher needs to take relevant steps to prevent the potentially hazardous situation. Lack of Train Integrity information has a significant impact on the performance of the line.

Performance indicators:

- Loss of integrity duration - duration that the train had lost its integrity (performance, quantitative);
- Probability of train integrity loss - probability that the train integrity is lost (safety, quantitative).

- OS#7 - Shunting Movement

Abstract:

ETCS includes a mode called shunting (SH), which enables trains to be moved both forwards and backwards and without the need for the trackside to issue movement authorities. Having granted permission for the train to enter SH, the trackside has very restricted functionality available to manage the train movement or to restrict it from entering an operational line leading to collision.

Description:

This operational scenario assumes that the ETCS Level 3 moving block is able to manage a possible driver's request for shunting anywhere on the line, but could decide to reject this and restrict shunting to predefined shunting areas. We describe two variants, one that considers the train entering the temporary shunting area manually, and the other one entering the same area automatically.

Performance indicators:

- Average time to resume normal driving - average time for the train to cross the shunting area (performance, quantitative);
- Probability of unauthorized exit from shunting area (safety, quantitative).

- OS#8 - End of Mission

Abstract:

This scenario describes the End of Mission (EoM) process for an L3 Area.

Description:

When a train completes a journey and the Driver closes the desk, the onboard issues an EoM request, and the train disconnects.

Performance indicators:

- Completeness - the probability that an object-status remains unknown/null (reliability & safety, quantitative);
- Reliability - the probability that a vital object state value is wrong (reliability, quantitative).

- OS#9 - Supervising Distance in Normal VCTS Driving

Abstract:

Supervision of train separation of a Virtual Coupled Train Set (VCTS) during normal driving.

Description:

This operational scenario addresses the supervision of train separation distance during normal driving in Virtual Coupling, and specifically, it assumes that Virtual Coupling has been already initiated.

The scenario starts with a VCTS (made of at least two trains) running under nominal

Virtual Coupling conditions and aims at evaluating VCTS system safety and performance.

Performance indicators:

- Probability of hazards due to positioning or communication faults – Probability of having an incorrect safe distance due to positioning errors or delay/errors/loss of communication (safety, quantitative)
- Line capacity – Measuring the expected increase of the line capacity compared to non-VCTS (performance, quantitative)
- OS#10 - Splitting of a VCTS Initiated by Slave

Abstract:

Termination of a Virtual Coupling session by splitting of a Virtual Coupled Train Set (VCTS) initiated by a slave.

Description:

This operational scenario addresses the termination of a Virtual Coupling session, and in particular the splitting of a VCTS initiated by a slave.

The scenario starts with a VCTS running under normal Virtual Coupling driving and ends with two VCTSs (possibly two standalone trains) running under moving block signalling.

Performance indicators:

- Probability of collision – probability that the relative distance between two trains in a virtual coupled train set becomes zero or less (safety, quantitative)
- Splitting time – the minimum time it takes for a slave train to split from a virtually coupled train set (performance, quantitative)

3.2.2 Template for operational scenario detailed description

In this subsection, the template describing the OSs is presented. The template is in the form of a word document composed of twelve tables. This section reports the tables separately and describes the meaning of the data contained in them for the sake of simplicity. In the Appendix the template is populated with respective OSs data.

Operational Scenario #X	
Title:	
Abstract:	
Description:	

The first section of the template concerns the general information about the OS, reporting its *title*, an *abstract*, i.e. a couple of lines clearly stating the context of the OS and its main objective, and a *description*, i.e. a longer text where more data are reported, especially on the conditions constraining the scenario.

Applicable Use Case(s)

1	
2	

This table reports the list of all the *Applicable Use Cases*, both for the moving block and for the virtual coupling, as they are described in deliverable D1.1.

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description

Performance indicators play an important role in the scope and industrial impact of the OS. Each performance indicator defines a qualitative/quantitative aspect of interest of the ETCS MB/VC system. In relation to the different parameters values and conditions the OS may be subject to, these indices may vary in their values determining if such conditions are acceptable or not according to requirements and specifications and/or represent a normal/degraded service level also on the basis of signalling type (high speed lines, high capacity lines, etc.). More in the details, the performance indicator, identified by a *name*, can be qualitative or quantitative (i.e., its *type*) and can be referred to one *property* (i.e., Logical, Functional, Availability, Reliability, Safety, Performance). Then, some performance indicators may be related to functional/non-functional requirements, prescribing a *threshold or a range* of acceptance for the performance indicator. A *description* concludes the definition of each performance indicator.

Signalling Type	System Type	Track Information
		(add a figure, if necessary)

The fourth table is intended to provide some context of the “test track”: in other words, notwithstanding the fact that the system behaviour specified should be valid in almost all track configuration, the OS is defined in a specific track configuration that represents the concrete setting (even if virtualised/simulated) in which the OS is analysed. This information considers: the *signalling type* (i.e., the kind of market segment the OS wants to address - freight, high speed lines, metro systems, etc.), the *system type* (i.e., full moving block, fixed virtual block, both with and without virtual coupling), and the *trackside information*, a high-level schema of the track (in terms of length of the line, presence of control points, switches, etc.).

Functional components

This table lists the *functional components* as reported in deliverable D1.1.

Trackside Function(s)	ETCS On-Board Function(s)

This table lists the functions of both *trackside* and *on-board* as reported in deliverable D1.1.

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)				
Train				
Speed				
Distance				
Communication				
Track				
Position				

This table defines the input parameters of the OS: parameters values may determine the qualitative and quantitative behaviour of the systems in a significant way. Hence, to accomplish a suitable analysis of the ETCS MB/VC system, choosing a proper set of these values is of a paramount importance. The first column of this table reports the macro-category of the parameter: Timers (whose timeout can determine different behaviours), Train (e.g., its length, weight, wheel adherence factors, etc.), speed (e.g., maximum allowed speed in a certain ETCS mode, initial speed of the train at starting of the OS), distance (e.g., length of a non-stopping/shunting area or initial distance between the head of the train and a switch), communication (e.g., latency of the radio communication media, probability of communication errors), track (e.g., friction factor of the tracks) and position (e.g., confidence of getting the correct position of the train by the GNSS equipment). Of course, other categories may be added. For each parameter, the following information should be provided: *name* (symbolic name of the parameter), *value/range* (admissible range in which the parameter could vary), *description* (explaining its meaning) and a *reference* to a standard and/or requirement.

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
Desc.				
	#1			
	...			
	#n			
A				
Desc.				
	#(n+1).A			
B				
Desc.				

#(n+1=. B	
...	

This table reports the evolution of the system in relation to the conditions described in the OS. Since usually the system may have different evolutions, according to the specification of the use cases on which the OS is based, the inclusion of alternative branches becomes necessary. A branch is one of the possible sequences of steps the system may follow, depending on one or more conditions that might occur. The occurrence of such conditions is determined by the values of the parameters; as an example, let us consider the two situations that can occur in case of (1) perfect communication network (i.e., where it is not possible to lose the communication between trackside and on-board) and (2) imperfect communication (i.e., where the communication reliability is determined by a probability value). Each branch is reported in the following lines.

Branch	Pre-conditions	Post-conditions	Triggers	Invariants/Assertions/...
<<bid>>				
Desc.				

An identifier of the branch is reported in the first column (i.e., the <<bid>> tag). As a convention, we do not name the main branch (the behaviour of the system before the occurrence of any branching conditions); the other branches may be denoted by capital Latin letters. On the right of the <<bid>>, the following information is reported: *pre-conditions* (describing the conditions the system fulfils before the “execution” of the branch), *post-conditions* (the conditions the system fulfils after exiting the branch), *triggers* (special events initiating the branch) and *invariants/assertions* (conditions that are fulfilled by the system across the entire sequence of events characterizing the branch). In the cell at the right of the *Desc.* label, a small description of the condition originating the branch is reported. Once the general data of each branches are described, a sequence of steps is listed, which includes the actions of the system under the considered branch.

Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
Desc.				
	#1			
	...			
	#n			

The steps (i.e., the white table rows) are simply characterized by a progressive number and a description of the action the system performs. There are two considerations to remark:

- the numbering of the steps is progressive, hence if an OS has a branch, and the main branch ends with the step #n, the first step of the branch A should be named #(n+1).A, the first of the branch B, #(n+1).B, and so on;
- the methodology used in this deliverable to describe the OSs is not formal: the objective is not to describe behaviours and conditions in a machine-interpretable format but to explain them in a human-readable unambiguous way: hence, the steps and the conditions will not be described according to a formal syntax.

Variant	Description	Alternatives	Main case	Impact/Affected Steps

Variants characterised the OS by providing possible alternatives that may affect the value of the performance indicators, according to a specific set of parameter values, even though they do not generally change the behaviour of the system. The table reported above contains information about such variants: *variant* is a symbolic name for the variant (e.g., V1); *description* explains it; *alternatives* reports the different alternatives of the variant (also denoting it with a small identification as V1.I, V1.II, etc.); *main case* indicates which one of among the possible alternatives is considered in the first instance in the OS; and *impact/affected steps* reports the steps of the OS's behaviour that are impacted by the variants (a clear reference may be substituted by a description for the sake of the simplicity).

Hazards		
ID	Description	Reference/new possible hazard

This table reports the hazards that are related to the OS: *id* is hazard identification, as they are reported in the source documents (e.g., [D3.2_X2Rail-2]), then a *description* and the *reference* to applicable documents are provided.

ID	Applicable Operational Rules	Reference

This tables contains the list of the operational rules applicable to the OS. For each operational rule, the following information is reported: an identifier (*ID*), equal to the one reported in the source document, if present; a description (*Applicable Operational Rules*) and the reference to the source document (*Reference*).

ID	Applicable Requirements	Reference

The same structure applies to referenced requirements: an identifier (*ID*), a description (*Applicable Requirement*) and the reference to the source document (*Reference*).

The description of the operational scenarios in the form given by this template is reported in Appendixes from A to J.

4. Methodology for System Modelling and Analysis

In this section, we first present the generic workflow comprising the iterative steps one can perform in order to produce formal models of moving block (MB) and virtual coupling (VC) systems, systematically. The workflow includes the verification and validation of formal models also, as the last step that delivers the final results. Next, the objective of each step of the workflow is detailed in a corresponding subsection.

4.1 Workflow

This subsection is devoted to the description of a workflow for the tasks to be undertaken within WP2 focused on the development of formal models. A workflow can be presented as a succession of steps and exhibits the inputs/outputs of each step. It aims to provide a methodological process to establish formal models for the MB system. Figure 5 presents a high-level view of the workflow. The items in bold in the figure are the **outputs** of the workflow. The details and inputs/outputs of each step will be provided in the next sub-sections.

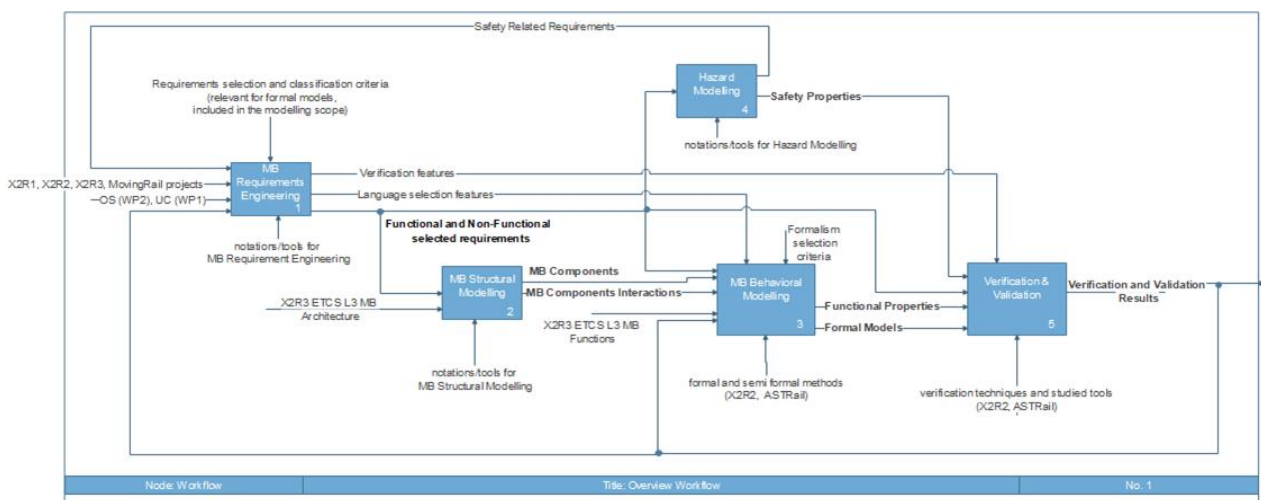


Figure 5. Workflow structure.

4.1.1 MB Requirements Engineering

The first step is MB Requirement Engineering. Its objective is to identify, starting from Operational Scenarios (OS-WP2) and Use Cases (UC-WP1), the most important features to consider in the modelling methodology and to identify and classify the most relevant requirements for MB system. For this step, the inputs are the description of OS and UC, the outcomes of X2Rail-1, X2Rail-2, X2Rail-3 and MovingRail, as well as the outputs of the Hazard Modelling (Safety Related Requirements) and of the Verification and Validation (Verification and Validation Results) steps from previous iterations of the modelling methodology. The **outputs** are a set of identified features to be considered in the modelling methodology and a list of selected functional and non-functional requirements.

MB Requirements Engineering can be decomposed in four sub-steps: 1) Analysis of Operational Scenarios and Use Cases, 2) MB Requirements Selection, 3) MB Requirements Refinement and 4) MB Requirements Modelling and Classification. The detailed structure of the MB Requirements Engineering phase is described in the Figure 6.

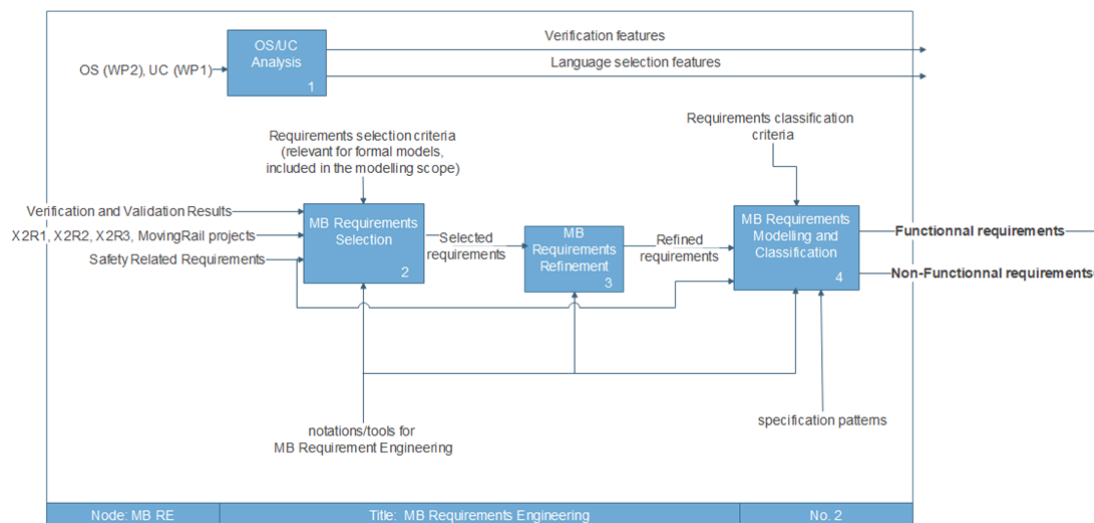


Figure 6. MB Requirement Engineering structure.

The first sub-step pertains to the Operational Scenarios and Use Cases Analysis. Its objective is to identify the most important features to consider in the modelling methodology. For this sub-step, the **input** is the description of selected Operational scenarios and Use cases. The **output** is a set of identifying features to be considered in the modelling methodology.

Starting from the analysis of the Use Case behaviour, as well as the UC and OS **parameters**, it is important to identify the particular features to consider (ex. temporal aspects, etc.). To represent **value/range of parameters**, the selected language should support complex data structure or abstraction method should be employed. Based on the **performance indicators**, different types of properties can be considered (safety, availability, qualitative, quantitative). For most of them, either a time related property is needed or a probability property. For **Language selection**, required features may be the ability to express temporal aspects, probability and complex data structure. For **Verification**, features may include analysis of time related properties and probability properties.

The **objective** of the MB Requirements Selection sub-step is to identify the most relevant requirements for MB system. For this sub-step, the **input** are the outcomes of X2Rail-1, X2Rail-2, X2Rail-3 and MovingRail projects, the output of Verification and Validation step (Verification and Validation Results) and the output of Hazard Modelling step (Safety Related Requirements). The **output** is a subset of requirements according to **some selection criteria** (e.g., based on the relevance of the requirement for the development of the formal model, according to the considered modelling scope).

The **objective** of MB Requirements Refinement is to refine the selected requirements. For this sub-step, the **input** corresponds to the output of MB Requirements Selection sub-step and the

output is a set of refined requirements.

The **objective** of MB Requirements Modelling and Classification is to classify and model the refined selected requirements. For this sub-step, the **inputs** are the output of MB Requirements Selection sub-step and the output of Hazard Modelling step (Safety Related Requirements). The **output** corresponds to a classification of requirements. Requirements can be classified into “**functional**” and “**non-functional**” requirements.

For MB Requirement Modelling, standard notation such as, for instance, SysML requirement diagrams, can be used.

4.1.2 MB Structural Modelling

The **objective** of the MB Structural Modelling is to identify the different components included in the MB System. The **inputs** are X2R3 ETCS L3 architecture and the output of MB requirements classification (the selected functional and non-functional requirements). The **output** corresponds to the different MB components and the interaction between them.

The detailed structure of MB Structural Modelling step is described in Figure 7.

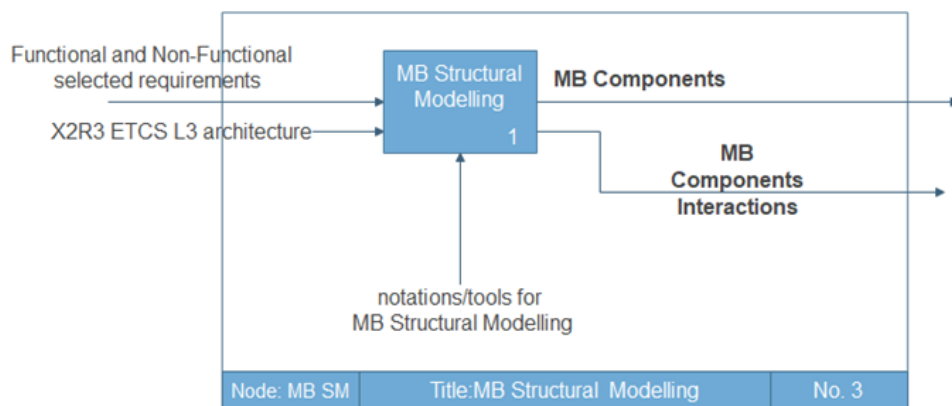


Figure 7. MB Structural Modelling step structure.

For MB Structural Modelling, it is possible to use some standard notations such as, for instance, the Class Diagram of UML / Block Definition Diagram of SysML and/or Component Diagram of UML/Package Diagram of SysML.

4.1.3 MB Behavioural Modelling

The objectives of MB Behavioural Modelling are 1) the identification of the main functions of MB system, 2) the assignment of a set of functions to each component, 3) the assignment of a set of functions to the interactions between identified components, 4) the selection of modelling formalism 5) the development of a parameterizable formal model for each component 6) the aggregation of models while considering the interactions between components and 7) the

identification of a set of functional properties. The inputs are X2R3 ETCS L3 MB functions, the output of MB requirements classification (the selected functional and non-functional requirements), the output of previous Verification and Validation steps (Verification Results) and the output of the MB Structural Modelling step (MB Components and MB Components Interactions). The outputs correspond to a set of parameterizable Formal Components Models, Formal Global Model and the set of Functional Properties. The detailed structure of MB Behavioural Modelling step is described in Figure 8.

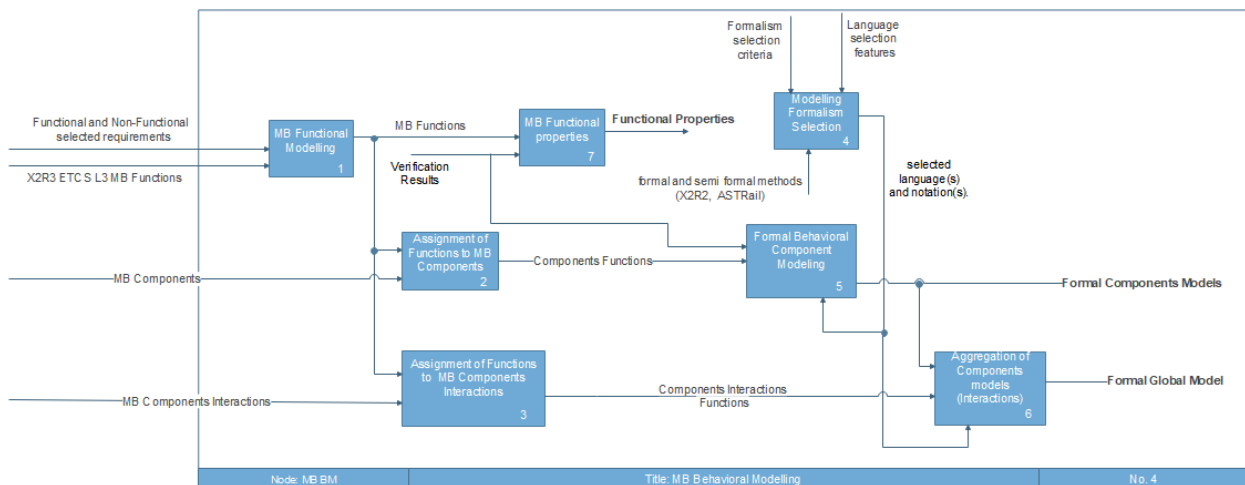


Figure 8. MB Behavioural Modelling step structure.

Regarding the sub-step MB Functional Modelling, the **objective** is to identify the main MB functions. It takes as **inputs** functional and non-functional requirements and X2R3 ETCS L3 MB Trackside Functions. It provides as **output** the set of main MB functions.

For the sub-step Assignment of Functions to MB Components, the **objective** is to assign a set of functions to each component. It takes as **inputs** the output of MB Functional Modelling sub-step (MB Functions) and the output of the MB Structural Modelling step (MB Components). It provides as **output** the Components Functions.

For the sub-step Assignment of Functions to MB Components Interactions, the **objective** is to assign to interactions between identified components a set of functions. It takes as **inputs** the output of MB Functional Modelling sub-step (MB Functions) and the output of the MB Structural Modelling step (MB Components Interactions). It provides as **output** Components Interactions Functions.

For the sub-step Modelling Formalism Selection, the **objective** is to select the modelling formalism for the design of formal models. The set of formal and semi-formal methods investigated in X2R2 and ASTRail projects can serve as a starting point. In addition, the output of the operational scenarios analysis (Language selection features) can guide the selection process. In general, the **formalism selection criteria** may include the support of temporal aspects, probability, data structure, parametrization, modularity, and concurrency. Language flexibility (import/export) and supported tools are also important for the selection. The **output** of this sub-step corresponds to a set of selected language(s) and notation(s).

For the sub-step Formal Behavioural Component Modelling, the **objective** is to develop parameterizable formal models for the MB components. It takes as **inputs** the output of the sub-step Assignment of Functions to MB Components (Components Functions) and the output of Verification and Validation step (Verification Results). Using the selected language(s) and notation(s), it produces as **output** the Formal Components Models.

For the sub-step Aggregation of Components models, the **objective** is to develop a formal global model including the interactions between the components. It takes as **inputs** the output of the sub-step Assignment of Functions to MB Components Interactions (Components Interactions Functions) and the output of the sub-step Formal Behavioural Component Modelling (Formal Components Models). Using the selected language(s) and notation(s), it produces as **output** the Formal Global Model for some considered context/scope.

The reusability aspect is ensured thanks to the fact that the component models should be parameterizable and also in the sub-step Aggregation of Components models where components models designed in the previous sub-step are used to form a global formal model considering the interactions among components.

For the sub-step MB Functional properties, the **objective** is to identify a set of functional properties. It takes as **inputs** the output of MB Functional Modelling (MB Functions) and the output of Verification and Validation step (Verification Results). It produces as **output** a set of functional properties.

Properties can be expressed in temporal logical languages such as LTL (Linear Temporal Logic), CTL (Computation Tree Logic), TCTL (Timed CTL), PTCTL (Probabilistic Timed CTL). Different types of properties may be defined such as:

- Invariants: An invariant specifies that all states of the system must respect some given property.
- Safety properties: A safety property states that no undesirable fact from the safety point of view should happen. A safety property can be an invariant, but could also specify conditions on the execution of the system.
- Deadlock freedom: This property states that the system will never lock up, i.e. there is no state in which the system can no longer evolve. This is a specific safety property that is generally very useful for system verification.
- Liveness properties: Intuitively, a liveness property specifies that the system can always reach some desirable situation (something good) will happen in the future.
- Reachability properties: A reachability property defines whether some state respecting some specific condition can be reached or not.

4.1.4 Verification & Validation

The Verification & Validation step aims to verify and validate the produced formal models. It takes as **inputs** the output of MB Behavioural Modelling (functional properties, Formal Components Models, and Formal Global Model), the output of Hazard Analysis and the output of

MB requirements classification sub-step (the selected functional and non-functional requirements). The **output** is the results of the verification and the validation.

Regarding the verification, possible techniques are:

- **Model checking** techniques consist in systematically exploring the state space of a formal model of a system to verify that the possible model evolutions satisfy the desired requirements expressed in the form of suitable temporal properties, such as, e.g., no collisions among trains, or no derailments, ever occur.
- **Theorem proving** is a subfield of automated reasoning, where verification reduces to proving the validity of logical formulae, describing both the system dynamics and the requirements, by means of automatic or semi-automatic (i.e., interactive) deduction techniques.

Verification approaches may take advantage of techniques for automatic model generation, such as:

- **Model transformation** can be seen as the automatic generation of a target model starting from a source model based on some transformation function.
- **Refinement** consists in applying a rigorous refinement process to a given abstract design in order to obtain a more concrete implementation of that design.
- **Refinement verification** is methodology of verifying that the functionalities of an abstract system model are correctly implemented by a lower level implementation.

The **validation** is an essential step to ensure that the obtained models produce the expected results. The Validation can be performed by generating tests to be applied either on the model itself, or on the implementation generated from the model. The Validation environment can then correspond to a simulation environment or to a real environment.

The Verification and Validation step can be decomposed into a set of sub-steps. Figure 9 presents the structure of this step.

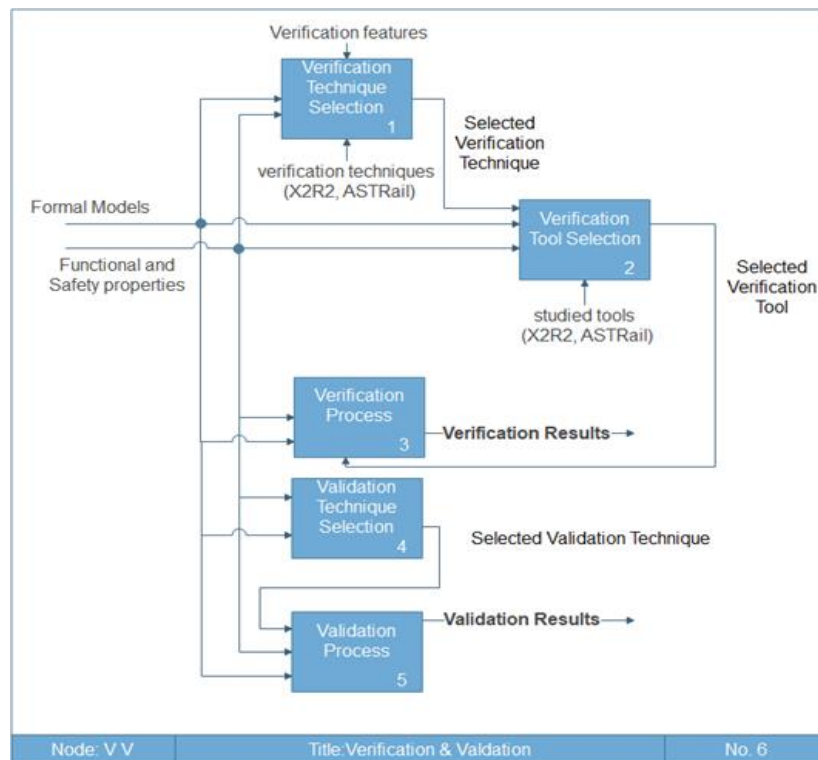


Figure 9. Verification and Validation step structure.

The first sub-step is the Verification Technique Selection. Its **objective** is to select the most suitable verification technique. It takes as **inputs** the formal models, the defined properties (functional and safety). Using the verification techniques enumerated in the surveys realized in X2R2 and ASTRail projects, it selects one or more Verification Techniques. The selection has to take into account the type of the developed formal models as well as the properties to be verified.

The **objective** of Verification Tool Selection sub-step is to select the verification tool(s) that support the selected verification technique, the selected modelling formalism and the selected properties. It takes as **inputs** the selected verification technique, the formal models, the desired properties (functional and safety) and selects a verification tool(s) among those studied and published in the X2R2 and ASTRail projects.

The **objective** of the Verification Process sub-step is to determine if the designed formal models satisfy the desired properties, by using the selected verification tool. It takes as **inputs** the formal models, the desired properties (functional and safety). The **output** is the results of verification.

The **objective** of Validation Technique Selection sub-step is to select the most suitable validation technique. It takes as **inputs** the formal models, the desired properties (functional and safety).

The **objective** of the Validation process sub-step is to validate that the designed models produces the expected results. It takes as **inputs** the selected validation technique, the formal models, the desired properties (functional and safety).

4.1.5 Hazard Modelling

The Hazard Modelling step aims to model the identified hazards that are related to the MB system. It takes as **inputs** the output of MB requirements classification sub-step (the selected functional and non-functional requirements). The **outputs** are Safety Related Requirements and Safety Properties. Figure 10 illustrates the structure of this step.

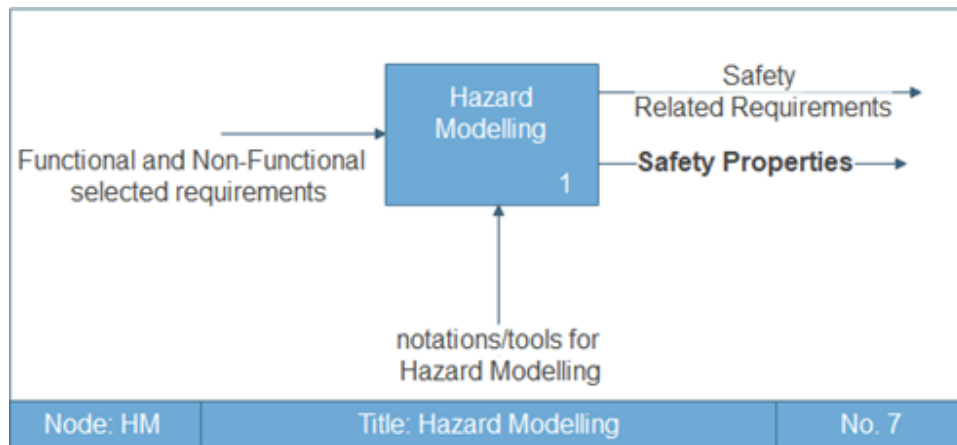


Figure 10. Hazard Modelling.

4.2 Modelling Principles and Artefacts

So far, we have presented a structured process for the systematic development of moving block models. The described workflow is based on several **modelling principles** and encompasses many **activities** that will be supported by **tools** and produce or depend on a number of **artefacts**. These aspects, that have only been mentioned in the general presentation of the modelling process, are discussed here in more details. In doing so, the role played by the artefacts and their management is also illustrated.

Modelling principles:

1. **Different levels of modelling.** It is evident from the modelling process definition that models are developed for different purposes and according to different perspectives on the system and abstraction levels. Hence, the modelling activities will first use *engineering modelling languages*, such as UML and its extensions (e.g., SysML) and, then, behavioural *formal models* expressed in formal specification languages (e.g., Automata, Petri Nets) will be developed to analyse properties and evaluate performance indices.
2. **Model-driven approach.** Model transformations can be defined, and possibly implemented in order to enable: a) the automated generation of formal models from other models (M2M transformations), b) the automated generation of textual artefacts

from models, for example system configurations or textual representations of models (M2T transformations).

3. **Model reusability.** Compositional modelling will be exploited as much as possible in order to support the definition of a *library* of models. To this end, models will be designed with clear interfaces. Template models, that is models parametric in some of their elements, may also be developed, so as to provide families of models that can be instantiated by choosing different values of their parameters. For example, a parametric model can be instantiated to describe the behaviour of different configurations of the same system. Model composability is not trivial to obtain, as formal models cannot be composed without considering that properties that hold for a sub-model (e.g., representing a system component) may not hold for the global model obtained by composing different sub-models. In addition, in case the sub-models are not written in the same formal specification language, the semantics of the composition and suitable composition operators must be defined.

The modelling principles described above will be applied whenever possible throughout the modelling process and help identify the nature of the artefacts that will be produced. Figure 11 shows the main layers on which the modelling process is based.

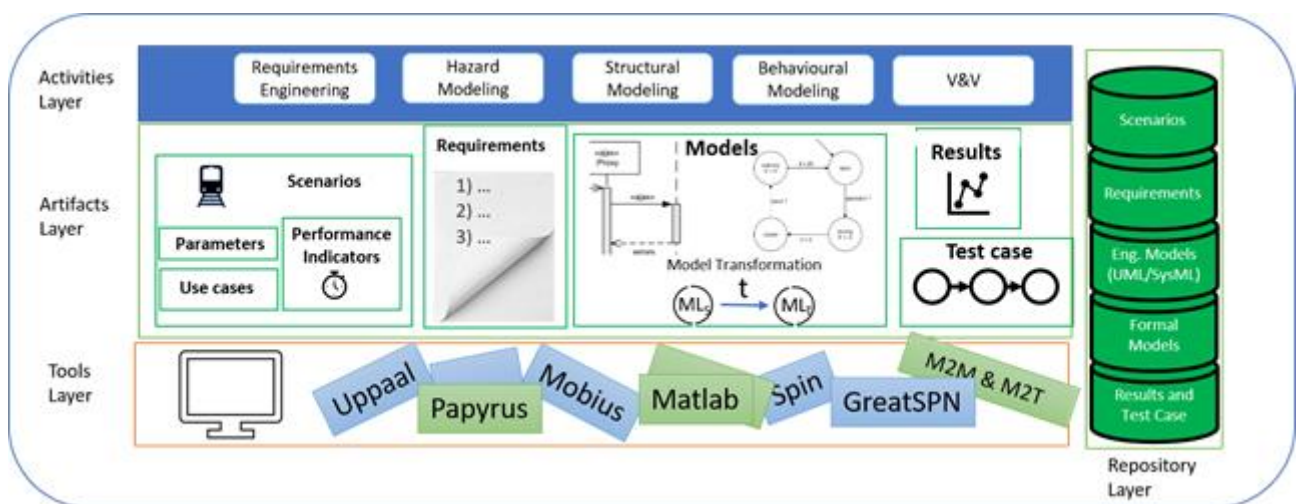


Figure 11. Modelling process layers.

All the modelling and evaluation tasks described in the previous sections are performed at the Activities Layer. The activities are possibly supported by tools of the Tools Layer and produce outputs or need inputs that are encoded by Artifacts. Artifacts, in turn, can be generated or used by tools and must be stored and managed at the Repository Layer.

The main classes of Artifacts are: Scenarios, Requirements, Models, Models Transformations, Results and Test Cases. Scenarios are built over use cases and include the parameters to be considered in the model definitions and the performance indicators to be evaluated. Requirements can be expressed in several ways, as already explained. Models are the primary artefacts in this context, they may be expressed by engineering modelling languages (e.g., UML/SysML diagrams) or using formal specification languages. Model Transformations are

artefacts, as well. They may be implemented and their implementation extends the set of available tools at the Tool Layer (“M2M & M2T” in Figure 11). The evaluation of performance indicators produces Results. Test cases could also be considered as a special case of Results, as they can be generated automatically from models.

Artifacts must be stored and managed to provide the basis for the development of an *integrated framework for Moving Block modelling and analysis*, which *should guarantee the traceability of relevant information* along the chain of activities and tools (e.g., requirements traceability, and the link between results and models). The framework could push the management of the artefacts a step further, and also *includes the management of tools*, by handling both artefacts and tools as specific *type of resources*, and being *integrated with the lifecycle applications used in the product development* (e.g., requirement management, quality management, etc.). Of course, *the definition and development of the modelling and analysis framework is out of the scope of this project*.

With the aim of providing a library of models, available hosting services will be considered, such as **SourceForge or Github**, that allow **to build and manage a repository with a number of different implementations of models and modelling solutions**. Although Github is mainly intended as a collaborative tool for software projects, it also offers the possibility to set up data **repositories**.

5. Guidelines for System Modelling

5.1 Requirements Modelling

This subsection is devoted to presenting the most relevant modelling approaches for requirement modelling.

According to [Soares_2011], there are several approaches to modelling requirements: Natural Language (NL), Structured Natural Language (Structured NL), User Stories, UML Use Case diagram, SysML Use Case diagram and SysML Requirements diagram. Basically, these approaches can be classified as purely textual graphics-based, or a combination of both.

The most common approach is to write requirements using natural language. With the purpose of giving more structure to requirements, structured natural language is used. User Stories (part of the eXtreme Programming agile methodology) can be written by the customer using non-technical terminology in a defined format of sentences using natural language. Use Cases are applied mainly to model functional requirements and are not very helpful for other types of requirements, such as non-functional ones. The SysML Use Case diagram is derived from the UML Use Case diagram with no significant changes. The SysML Requirements diagram is a standard way to model functional and non-functional requirements.

The workflow described in Section 4.1 is organized in a sequence of steps, with MB Requirement Engineering (MB RE) the first step. MB RE aims to identify and classify the most relevant requirements that apply to a MB system. It provides as output a set of functional and non-functional selected requirements written in Structured Natural Language.

In [Durugbo_2013], several requirements modelling criteria are reported:

- Human readable (the property of representations to be readable by humans),
- Requirement relationships (the presence of constructs to depict requirement relationships),
- Requirement types (the existence of features to illustrate types of requirements),
- Requirement prioritization (the ability of representations to prioritize requirements),
- Requirements grouping (ease with which related requirements can be grouped),
- Consistency (consistency in the stages or parts of the representation),
- Editable (support for editing during the elicitation process),
- Unambiguity (the use of well-defined semantics),
- Programmable (ease with which representations can be transformed into codes for computer programs),
- Correctness (the ability of representation to correctly depict interactions),
- Verifiable (ease with which requirements can be reviewed, inspected or tested),
- Traceable (ease with which the history of requirements can be ascertained).

The following table presents an evaluation of modelling requirement approaches according to the modelling criteria from [Durugbo_2013]. The symbols ✓, (✓), × mean that the modelling approach supports, supports partially, does not support the criteria, respectively.

	Natural Language	Structured NL	User Stories	UC UML	Req SysML
Graphical modelling	x	x	x	✓	✓
Human readable	✓	(✓)	✓	(✓)	(✓)
Requirement relationships	x	x	x	(✓)	✓
Requirement types	(✓)	(✓)	(✓)	x	✓
Requirement prioritization	x	(✓)	✓	x	x
Requirements grouping	✓	x	x	✓	✓
Consistency	x	x	x	(✓)	✓
Editable	(✓)	(✓)	(✓)	(✓)	✓
Unambiguity	x	(✓)	x	x	(✓)
Programmable	x	x	x	(✓)	(✓)
Correctness	(✓)	(✓)	(✓)	(✓)	(✓)
Verifiable	x	(✓)	x	(✓)	(✓)
Traceable	(✓)	(✓)	(✓)	(✓)	✓

It can be seen from the table that SysML requirement diagram covers most of the criteria. Let us provide a brief introduction to it.

A SysML Requirement diagram is a static structural diagram that captures hierarchies of requirements with additional relationships such as *derivation*, *satisfaction*, *verification* and *refinement*. A SysML Requirement diagram is formed by requirement elements and requirements relationships. Requirement elements are reported in [Abbors_2009] and requirement relationships are reported in [Soares_2011]. Each requirement element contains a *name* field, which specifies the name of the requirement, an *id* field, and a *text* field. The *id* field simply specifies an identifier of the requirement, whereas the *text* field describes the requirement. A requirement also contains a *source* field, which specifies the origins of the requirement. The relationships are *hierarchy* (represented by the symbol \oplus), *derive* (represented by the «stereotype» deriveReq), *master/slave* (indicated by the use of the copy «stereotype»), *satisfy* (represented by the «stereotype» satisfy), *verify* (represented by the «stereotype» verify), *refine* (represented by the keyword refine) and *trace* (represented by the «stereotype» trace).

Figure 12 shows an example of a SysML requirements diagram.

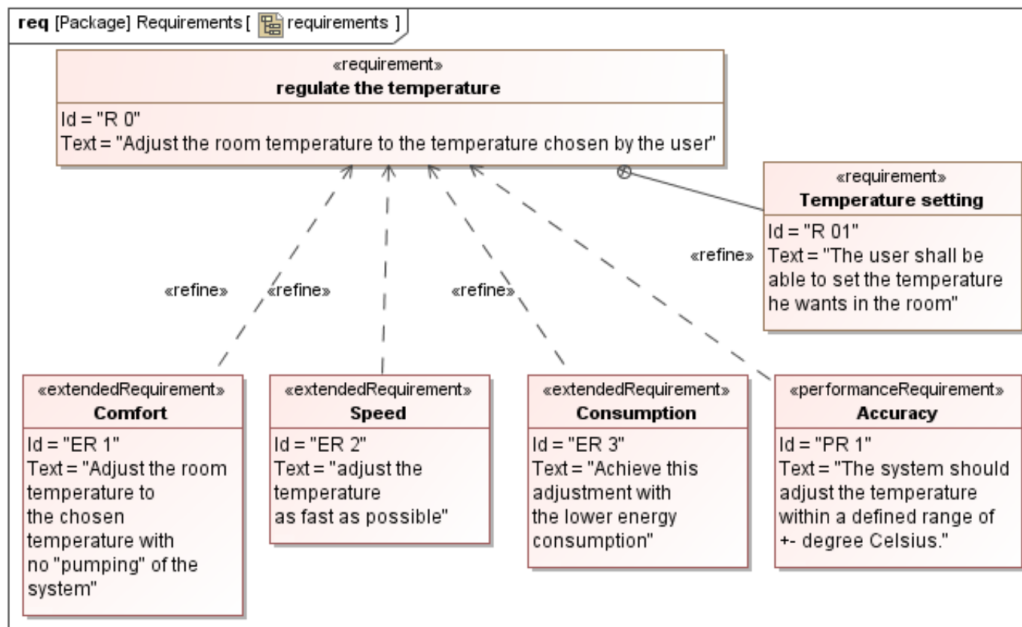


Figure 12. SysML Requirement Diagram toy example¹.

The objective of the workflow is the development of formal models for a MB system. In order to guarantee that all selected requirements in the requirement engineering step are considered in the designed formal models, traceability aspects should be considered. To this end, requirement processing tools such as rationale DOORS and POLARION may be employed. Requirement processing tools such as Rational DOORS and POLARION can be employed to trace requirements to the finished formal models making certain that all needs are fulfilled.

5.2 Structural Modelling

This paragraph explores the concept of structural modelling, starting from a brief review of definitions and approaches already present in the literature, as well as a description of the main languages used in the scientific literature and industrial practice. Domain modelling will be described by focusing on profiling techniques. The paragraph will end by reviewing recurrent modelling problems and possible solutions (patterns). Where useful, small examples will be included to show the applicability of the described concepts.

For structural modelling, we mean the set of techniques and languages designed to capture the decomposition of the entire systems into parts, their description in terms of property and interfaces they expose and the relationships between them. A similar definition is present in the scientific literature: "[a structural model] determines the required blocks and structures for each use case or requirement, and model all these in block diagrams" [Weilkiens_07].

Other software (and system) engineering approaches focuses more on system architecture. The

¹ <https://docs.nomagic.com/display/CRMP190/Requirement+Diagram>

concepts of structure model and system architecture differ in a meaningful way, since architecture is a broader notion that includes the structural model. This notwithstanding, there is some overlap between the two concepts, since one of the first step in system/software architecture design is the definition of a structural model.

As example:

- architecture is the fundamental organization of a software system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution [IEEE1471];
- architecture consists of the structure of the system combined with architecture characteristics the system must support, the architecture decision, and finally design principles [Richards_20];
- one of the pillars of the work of the SAF group [SAF] is the definition of a Logical Domain View (SLV) that focuses on considering the structure one of the most important -ilities (the architecture characteristics of the previous definition) of a system architecture;
- a set of fundamental concepts and properties in a specified environment, embodied in elements, relationships and principles to guide system design and evolution NOTE Systems architecture provides the conceptual definition of the logical, physical structures, behaviour, temporal relationships, and/or other aspects of a system and allocations among alternatives (e.g., physical elements, software, and operations; and/or functions in system-of-interest versus enabling system) [ISO/IEC/IEEE 42010];
- the structure (i.e., the parts that exhibit the behaviour, and the component hierarchy, elements, and stores) is a central part of the concept of system architecture in the INCOSE vision of Model Based System Engineering;
- the Unified Modelling Language traditionally structures its diagrams according to the 4 + 1 view schema [Kruchten_1995]. From the structural point of view there are three views where structural concerns can be found: Logical View (class diagrams, state diagrams), Development View (component diagram, package diagram) and Physical View (deployment diagram).

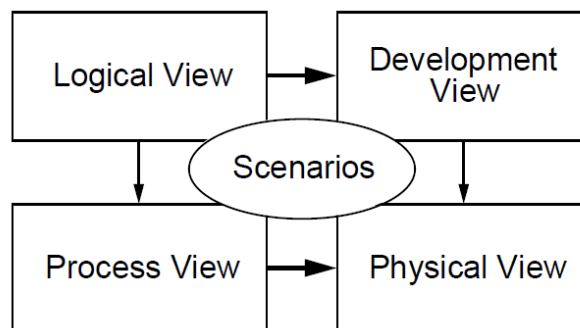


Figure 13. The 4 + 1 model schema [Kruchten95].

It seems evident from the cited approaches that the following concepts appear to be recurrent in structural modelling:

- hierarchical decomposition of the whole into parts,
- definition of blocks with clear interfaces,

- presence of information hiding mechanisms (separation of the interface from the implementation of a component/block/function),
- definition of the relationships between the parts,
- definition of the nature of the dependencies represented by these relationships (structural dependency, functional dependency, generalisation-specialisation dependency, etc.),
- definition of the flow of data between blocks.

From the perspective of the modelling languages, there are several approaches present in the literature and a complete survey of these languages is out of the scope of this document. This notwithstanding, in the following the most relevant approaches and related modelling formalisms (or part of formalisms) are briefly described.

UML

According to the UML superstructure [UML], UML diagrams represent structural aspects by means of different types of diagrams, as illustrated in Figure 14 (left side).

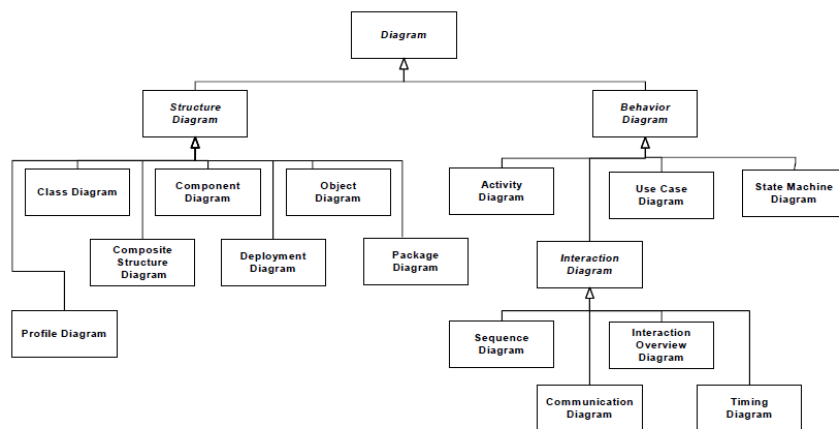


Figure 14. Taxonomy of the UML diagrams [UML02].

Class Diagram: mainly used to represents the relationships among the classes (i.e., abstraction of domain objects). Pragmatically, they should be used to represent abstract concepts (e.g., software items). According to the object-oriented paradigms UML relies on, classes can be characterized by properties, methods and can be related by means of static relationships, such as *generalization*, *association*, *aggregation*, and *composition*. Special classes, such as interfaces and abstract classes can be added to the diagram. Datatypes (enumeration in the example) and constraints (reported in the linked notes) can be added as well. Figure 15 shows a simple example.

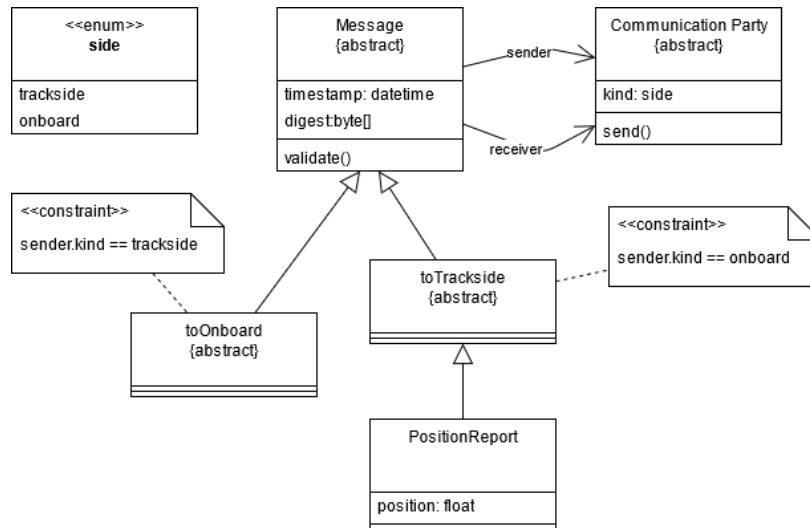


Figure 15. UML Class Diagram - example.

Object Diagram: object diagrams implement the relationships between the instances of classes: hence, this diagram is usually coupled with a class diagram. Figure 16 shows an example related to the one reported in Figure 15.

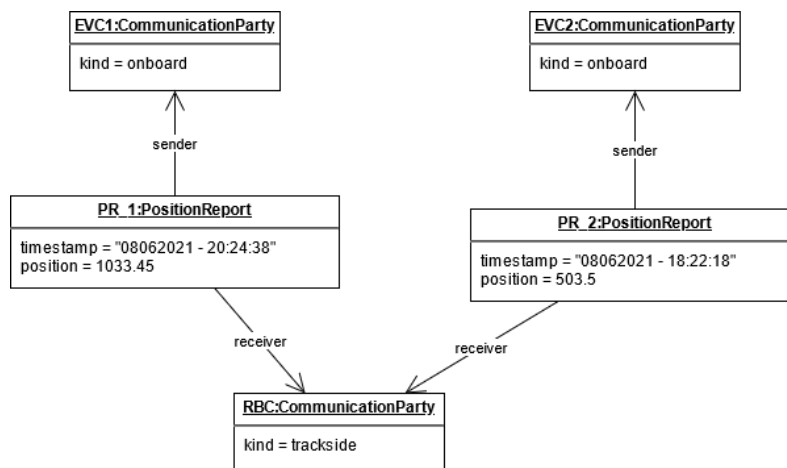


Figure 16. UML Object Diagram - example.

Component Diagrams: UML components differ from UML classes. A component is generally bigger and more abstract than a class. While a class is a relatively low-level blueprint of the software, a component might be a set of classes, which, taken together, form an encapsulated module offering a “service” that another component may require. Component diagrams do not show actual code, but, rather, the dependencies between components. Figure 17 reports a simple example, where circles represent services and arcs their usage.

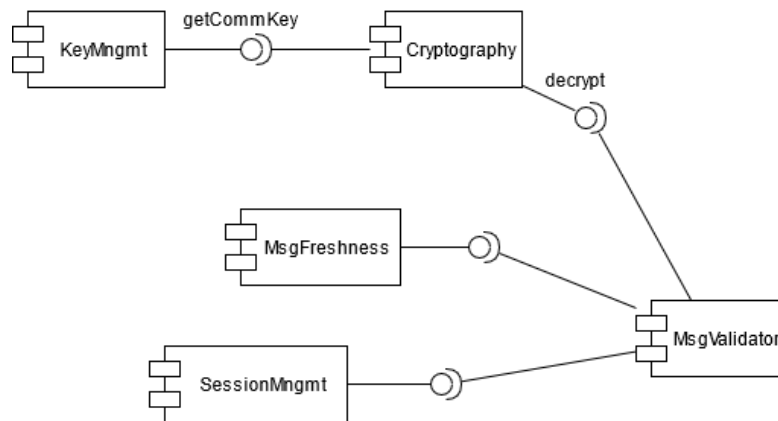


Figure 17. UML Component Diagram - example.

Package Diagram: packages are logical containers, where classes and other structural UML model elements can be grouped; packages can contain other packages. Packages are usually related by means of dependency relationships, which represents the fact that some elements contained in the package depend on the use of services or interfaces provided by some other elements in the package. Figure 18 reports a small example where it is clear that toTrackside depends on the Common package.

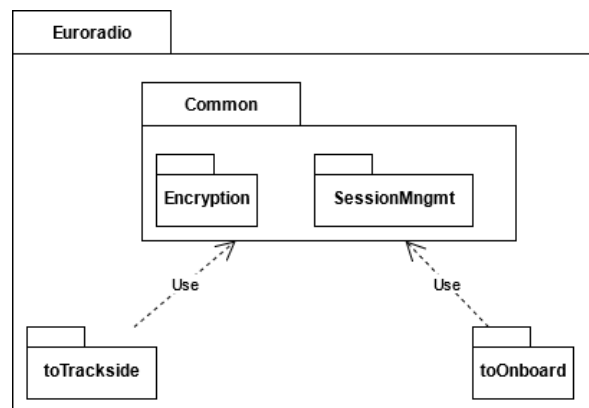


Figure 18. UML Package Diagram - example.

Composite Structure Diagrams: composite structure diagrams where not included in the first version of the UML language: they are part of the language since UML version 2.0, answering to the need of the modellers to investigate and specify components' internals. Figure 19 reports a small example which further details the Cryptography component contained in Figure 17.

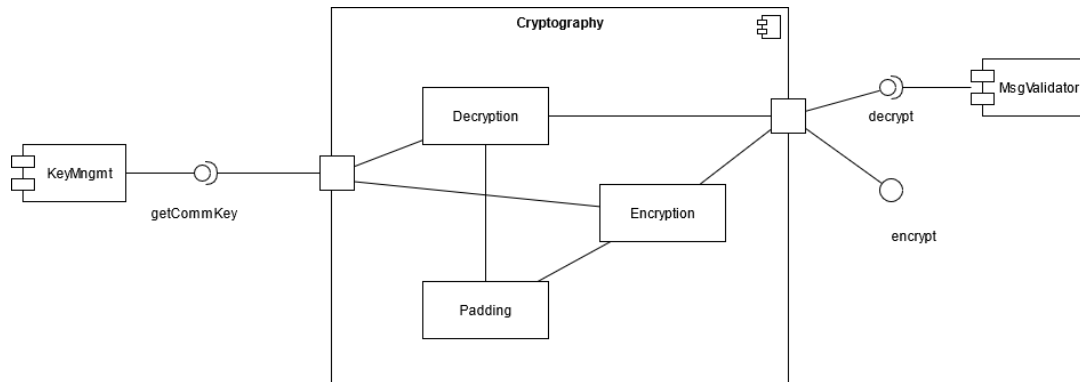


Figure 19. UML Composite Structure Diagram - example.

Deployment Diagram: they usually deal with the physical apportionment of the software infrastructure, defining hardware nodes and allocating software component to such nodes. Figure 20 reports a small example.

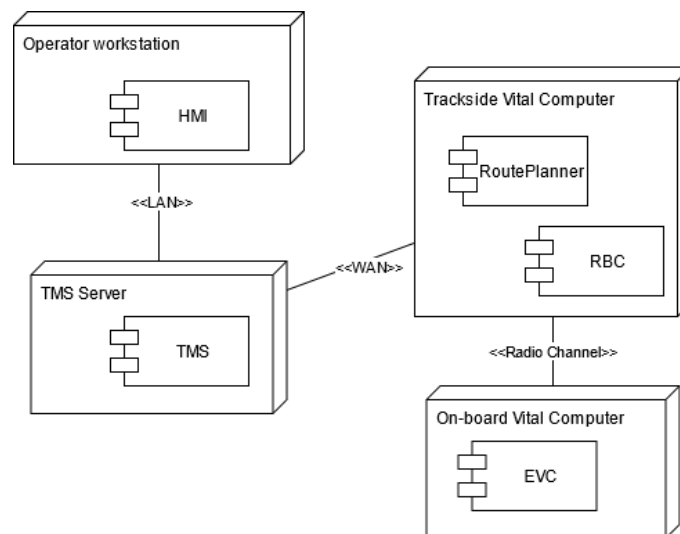


Figure 20. UML Deployment Diagram - example.

SysML Structural diagrams

OMG SysML includes diagrams that can be used to specify system requirements, behaviour, structure and parametric relationships. These are known as the four pillars of OMG SysML.

The OMG SysML diagram Taxonomy is shown in Figure 21 [Hause_2006].

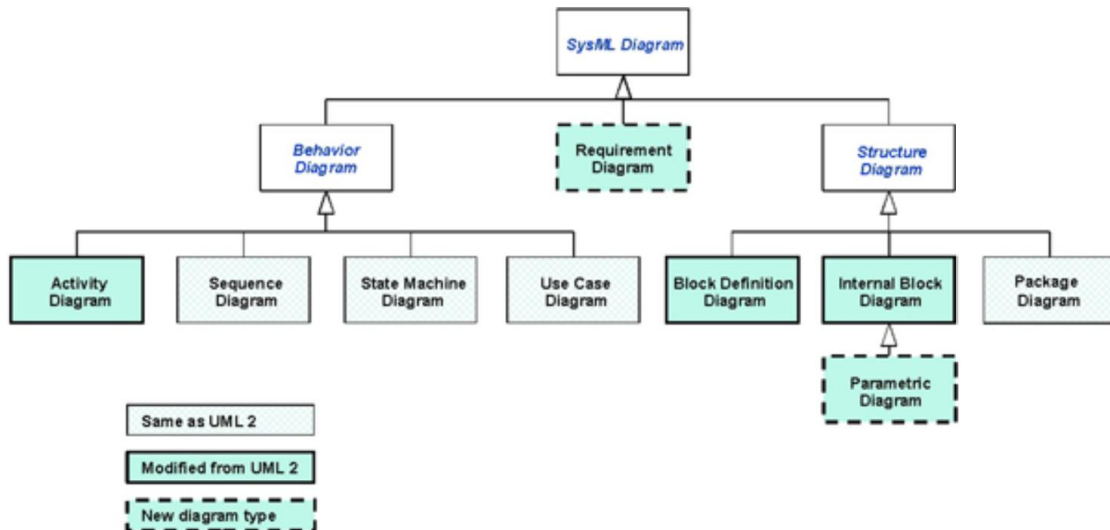


Figure 21. OMG SysML Diagram Taxonomy.

Each OMG SysML diagram has a *frame* with a *Contents* area, a *header* and a diagram description (see Figure 22). A detailed definition of diagram frame elements is provided in [OMG_2006] (Annex A).

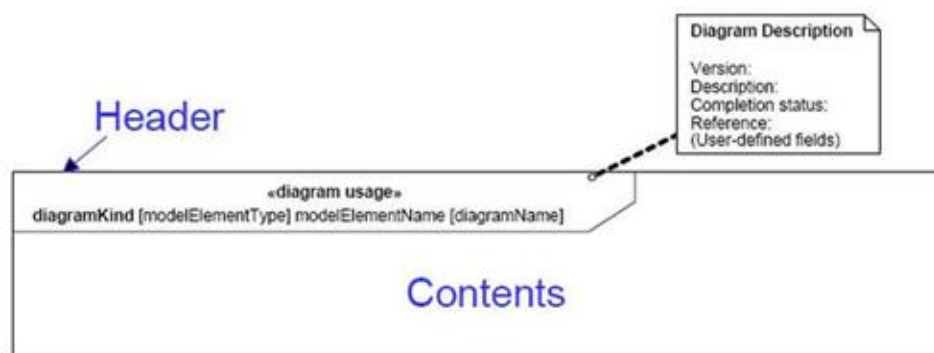


Figure 22. Diagram frame.

We focus in this subsection on SysML structure diagrams. Details are reported from [HAUSE_2006] and [OMG_2006]. The system structure is represented by *Block Definition Diagrams* (BDD) and *Internal Block Diagrams* (IBD). The BDD is based on UML structured class diagram. It describes the system hierarchy and system/component classifications. The IBD is based on UML2 composite structure diagram. It describes the internal structure of a system in terms of its parts, ports, and connectors. The package diagram is used to organize the model. The parametric diagram is a specialization of an IBD that represents constraints on system parameter values such as performance, reliability and mass properties to support engineering analysis.

The major structural extension in OMG SysML is the «block», which extends the UML Structured Class. Blocks provide a unifying concept to describe the structure of an element or system. They can represent any level of the system hierarchy, including the top-level system, a subsystem, or a logical or physical component of a system or environment.

In a BDD, a block is represented graphically by a rectangle organized into compartments. The name of the block appears at the top, and is the only required compartment.

All other compartments have labels indicating what they contain: values, parts, etc.

The stereotype "block" appears by default during modeling. Other keywords, such as "system" and "subsystem", are also available.

It is not mandatory to display within a block the attributes representing properties that characterize this block nor the operations that represent what can be requested from the block. In this case, the diagram carries little information, but it gives quick view of the structure of the system.

The example, presented in Figure 23, offers a basic structural modelling of a vehicle in the form of a "system block".

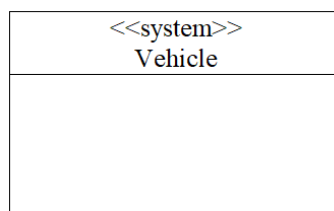


Figure 23. System Vehicle

It is possible to refine the block with a number of properties, such as:

- values that describe quantifiable characteristics,
- parts that describe the hierarchy of decomposition of the block in terms of other blocks,
- references that characterize the association between several blocks,
- constraints that characterize a condition relating to one or more elements of the model and that must be satisfied by the corresponding elements,
- operations that represent what can be asked of the block, having blocks also behavioural properties,
- ports that define the offered (provided) and required interaction points between blocks.

The example of the vehicle system, refined with a number of properties, is presented in Figure 24.

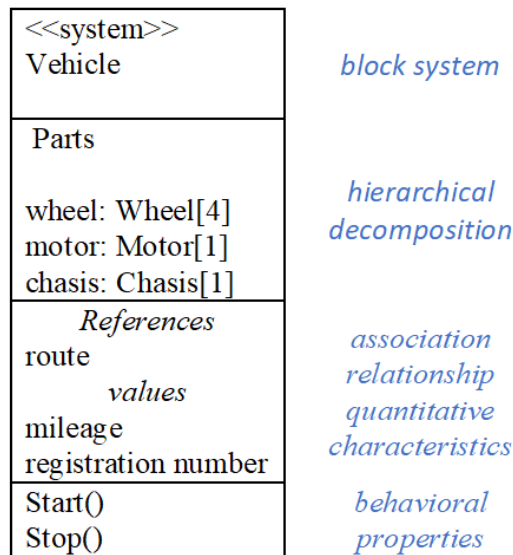


Figure 24. Vehicle block.

Each block can have a list of parts, references or value type.

An example of a block definition diagram, taken from [Friedenthal_2006], is provided in Figure 25.

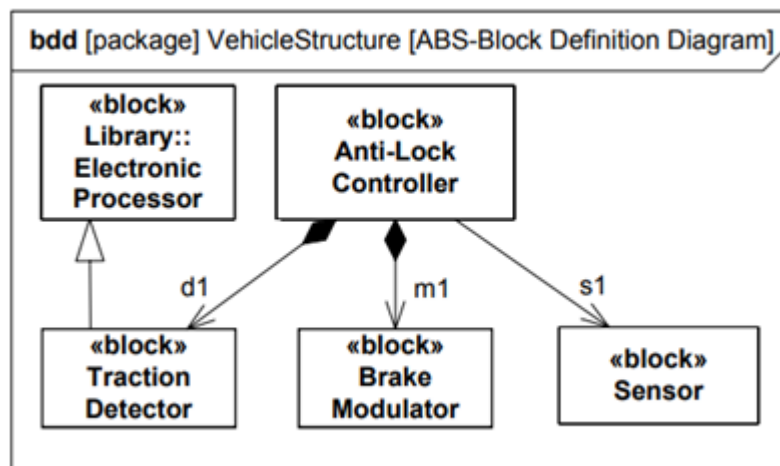


Figure 25. Vehicle Structure BDD.

The relationship between Anti-Lock Controller block and Traction Detector block is a composition relationship. The relationship between Anti-Lock Controller block and Sensor block is an association relationship and the relationship between Traction Detector block and Electronic Processor block is a generalization relationship.

Architecture Analysis and Design Language (AADL)

SEU of the Carnegie-Mellon University proposed its language for hardware/software highly coupled critical systems, named Architectural Design and Analysis Language (AADL), formerly known as Avionics Architecture Description Language since it was originally designed for the

avionics domain. We refer to the version 2 of the language. Due to its formal syntax and semantics, this language has been the study subject for researchers and practitioners who enriched the AADL ecosystems of several mappings to other formalisms and of concrete “AADL-to-X” transformation enabling MBSE. The set of AADL modelling features includes hierarchical decomposition, dependency, definition of component-level variables, separation of interface and implementation, signals, and ports. Behaviours and non-functional properties (e.g., timing properties) of components can also be added to an AADL model. Another important feature of AADL is the presence of proper linguistic constructs to capture error propagation concern. Figure 26 and Figure 27 report two samples of a larger AADL model related to the railway domain [Ex_AADL]: Figure 26 shows how a part can be described by its variables, state transitions, and error handling behaviour, while Figure 27 focuses of the overall system view.

```

device alarm
features
    alert : out data port boolean;
    reset : in event port;
annex behavior_specification {**
    states
        Inactive : initial state;
        Active : state;
    transitions
        t1 : Inactive      -[self.hazard_detected      ]-> Active;
        t2 : Active      -[reset                        ]-> Inactive;
        t3 : Active      -[                             ]-> Active {alert
:= true};
**};
annex EMV2 {**
    use types ErrorLibrary;
    use behavior train_errors::simple;

    error propagations
        alert          : out propagation{ValueError};
    flows
        f1              : error source alert{ValueError};
    end propagations;

    component error behavior
    events
        Failure : error event;
    transitions
        t0 : Operational -[Failure]-> Failed;
--  FIXME: how to show the synchronization between behavior and error model annex?
--  see below
--      t1 : Failed -[self.reset]-> Operational;
    propagations
        p10 : Failed -[]-> alert{ValueError};
    end component;
    properties
        EMV2::hazards =>
            ([
                crossreference => "N/A";
                failure => "Alarm is not operating";
                phases => ("");
                description => "";
                comment => "";
            ])
        applies to Failed;

        EMV2::hazards =>
            ([
                crossreference => "N/A";
                failure => "Alarm is fully operational";
                phases => ("");
                description => "";
                comment => "";
            ])
        applies to Operational;
    **};
end alarm;

```

Figure 26. AADL train part description

```

system train
end train;
system implementation train.il
subcomponents
    alarm      : device alarm;
    train_ctrl : system train_controller;
    door_ctrl  : system door_controller;
    door_sensor : device door_sensor;
connections
    conn_sensor : port door_sensor.value      -> door_ctrl.door_sensor;
    conn_alarm  : port alarm.alert             -> door_ctrl.alarm;
    conn_speed  : port train_ctrl.speed        -> door_ctrl.speed;
    conn_transit : port train_ctrl.intransit   -> door_ctrl.intransit;
annex EMV2 {**
    use types ErrorLibrary;
    use behavior train_errors::simple;

    composite error behavior
    states
        [alarm.Operational and door_ctrl.Operational and train_ctrl.Operational
and door_sensor.Operational]->Operational;
        [door_ctrl.Failed and door_sensor.Failed]->Failed;
        [alarm.Failed]->Failed;
    end composite;
**};
end train.il;

```

Figure 27. AADL train whole description.

SCADE

Another proprietary modelling languages and analysis framework is constituted by the SCADE suite. It is developed by ANSYS and counts a widespread adoption in many critical industrial contexts. The ANSYS tool ecosystem counts several tools spanning from embedded system design to multiphysics modelling and simulation. This clearly constitute a strong advantage of this suite with respect the others in case of hybrid systems. The SCADE suite heavily relies on the LUSTRE formal specification language which is an industrial-proven tool for specifying real-time critical system. Both graphical and textual notations are allowed. Figure 28 reports an example of modelling the interactions of the train (physics and control) with crossing system [SCADE].

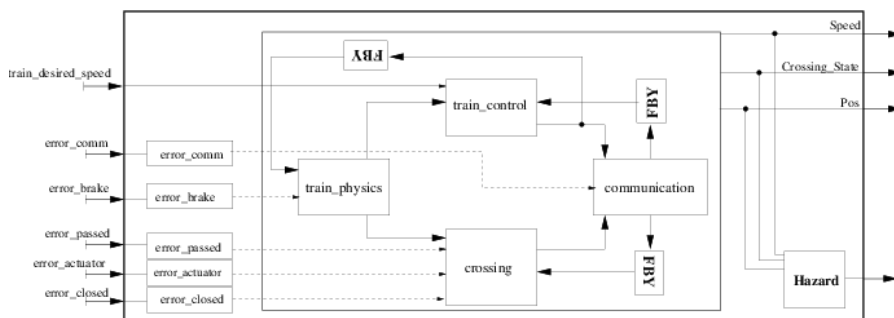


Figure 28. SCADE example [SCADE].

Simulink

The Matlab suite is developed by Mathworks and it represents a standard in many industrial domains (e.g., automotive where it is widely accepted in MBSE). The presence of certified compiler able to generate executable code from high level models have strongly fostered such an adoption. Simulink is the part of the Matlab framework more oriented to structural decomposition of a big system into hierarchical blocks that could be transparent (further specified by means of block decomposition) or opaque (characterized by a S-function or other behavioural specification methods). Figure 29 reports a model of train speed supervision system by means of the Simulink framework [SIMULINK].

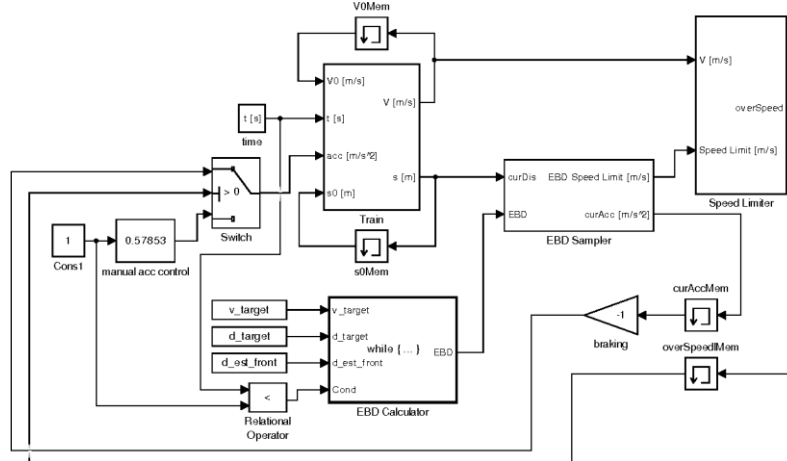


Figure 29. Simulink example [SIMULINK].

System Structure Modeling Language (S2ML)

The last modelling language to present for the structural modelling is the System Structure Modeling Language (S2ML), proposed in 2015 and focusing on structural parts [Batteux_15], developed in the framework of the OpenAltaRica project, led by IRT SystemX. Like the other approaches, S2ML supports composition, classes and inheritance, ports and connectors, aggregation and polymorphism. Concrete notation can textual as well as graphical. Figure 30 reports a small example of a textual description in S2ML.

```
block positionmanager
  port output;
  block gnss
    port position;
  end
  block imu
    port position;
  end
  block voter
    port pos1, pos2, pos;
  end
  connection [gnss.position, voter.pos1];
  connection [imu.position, voter.pos];
  connection [voter.pos, output];
end
```


Figure 30. S2ML example.

Often, “generic purposes” modelling languages do not cope with specific modelling needs. Domain specific concepts as well as cross-cutting concerns must be addressed to enable large diffusion of modelling approaches by enabling the annotation of extra-information in models. Among all the technical solutions to this problem, this document deals with the profiling techniques. UML Profile is a mechanism, introduced in UML2, to extend the UML language without redefining from scratch a whole modelling language (as it happens in other Domain Specific Modelling Language definition approaches). From a practical point of view, a profile diagram (referring to UML2) is a class diagram where elements are not classes but rather metaclasses (language elements of UML as classes, use cases, states, etc.) and stereotypes (“labels” that can be applied to model elements), tagged-values (structured typed data that add extra-information to a model) and constraints (OCL rules for supporting automatic validation of profiled models).

UML profiles may be of two types:

- “horizontal” profiles, that capture “abstract” aspects that are cross-cutting with respect to application domains (e.g., performance, persistence, etc.);
- “vertical” profiles, that report the “concrete” aspects of a domain (e.g., telecommunications, automotive, etc.).

MARTE is a well-known “horizontal” UML profile that focuses on performance and real-time aspects of a system [MARTE]. It is an OMG standard and is widespread in industry. MARTE-DAM specialises MARTE adding dependability modelling primitives. Some applications of MARTE-DAM in the railway sector are found in scientific literature [Bernardi_11, Bernardi_13]. For the railway “vertical” domain, Eulynx is an approach that is central in PerformingRAIL due to its capability to enable the RCA initiative of EU [Eulynx].

In an almost independent way from the modelling language, the end of this section is devoted to some recurrent modelling problems of system structures. These problems (and their solutions) are named in the scientific literature “patterns”. Patterns are known to the scientific community since the seminal work of [Alexander_1977] and widely diffused by the work of [GoF_1995]. In this last work, a subset of patterns is dedicated to structural patterns that, even if mainly related to software aspects, can be ported to the system engineering domain. Among the structural patterns:

- Bridge, that allows the separation between a class interface from its implementation;
- Composite structure, able to give an abstract method to manipulated tree-structured hierarchical items;
- Façade, proposed with the objective to deal with interfaces to different and heterogeneous system in a unified way.

5.3 Behavioural Modelling

For **behavioural modelling**, system dynamics can be reformulated by semi-formal and formal methods. From [D5.1_X2Rail-2] and [D4.1_ASTRAL], 27 languages/methods can be identified: UML, Labelled Transition System, State Machines, B/Event B, Timed Automata, Petri-Nets, SMV/NuSMV, CSP, Promela, SysML, UML state machines, SCADE, SDL, Message Sequence Charts, MARTE, Hybrid Automata/Stochastic Priced Timed Automata, PLC, SPES, OTHELLO, QTV, VDM, DSTM4Rail, Abstract State Machines, Lustre, PiSPEC, TLA and HLL.

This indicates that there is no clear, indisputable evidence or direction about which language/method to employ in railway systems development, and many languages/methods may be adequate for the same purpose.

5.3.1 Semi-formal Models

This subsection focuses on SysML behavioural diagrams, UML State Machine diagrams and UML Activity diagrams, which can be employed to specify the abstract behaviour of classes and instantiated objects for the MB and VC systems.

SysML behavioural diagrams

The details of the diagrams are reported from [Hause_2006] and [OMG_2006]. OMG SysML behaviour diagrams include use case diagram, activity diagram, sequence diagram and state machine diagram. A use-case diagram provides a high-level description of the system functionality. Activity diagram describe the flow of data and control across activities. Sequence diagram specifies the interaction among collaborating parts of a system. State machine diagrams describe the state transitions and actions that a system or its parts perform in response to events.

SysML Sequence Diagrams

Sequence diagrams describe the flow of control across actors and systems (blocks) or across different parts of a system. These diagrams specify the **messages sent and received** among the interacting entities by means of **lifelines**, where **time** is represented along the vertical axis. Sequence diagrams can describe highly complex interactions with special constructs for various types of control logic, reference interactions on other sequence diagrams, and decomposition of lifelines into their constituent parts.

Let us provide some **Basic Notations**.

Lifeline is a representation of the existence of a participating element in a sequence diagram. A lifeline has a *name* and a *type*. It is represented graphically by a vertical dotted line.

Message is a one-way communication element between lifelines that triggers activity in the recipient. The receipt of a message causes an event at the receiver end.

Activation of a lifeline is represented by vertical bands along a lifeline that specify the activation periods. These bands are optional, but allow to better identify the dotted arrow corresponding to the response message.

Dotted arrows represent feedback or response messages. This means that the message in question is the direct result of some previous message.

A synchronous message is represented by a solid arrow and an asynchronous by a hollow arrow.

Reflective messages are used to describe internal behaviour. Such messages are sent and received on the same lifeline level and graphically denoted by loops.

Combined fragments are logical groupings, represented by a rectangle, which contain the conditional structures that affect the flow of messages. A combined fragment contains interaction operands and is defined by the interaction operator². The type of combined fragment is determined by the interaction operator. For example:

- A “ref” interaction operator names a reference to a sequence diagram fragment defined elsewhere.
- A “par” interaction operator has 2 or more parts that execute concurrently (the order is undetermined).
- An “alt” interaction operator has 2 or more parts, but only one executes based on a condition/state. An operand fragment labelled [else] is executed if no other condition is satisfied.
- An opt [condition] interaction operator has 1 part that may be executed based on a condition/state value.

Figure 31 illustrates an example of a sequence diagram³.

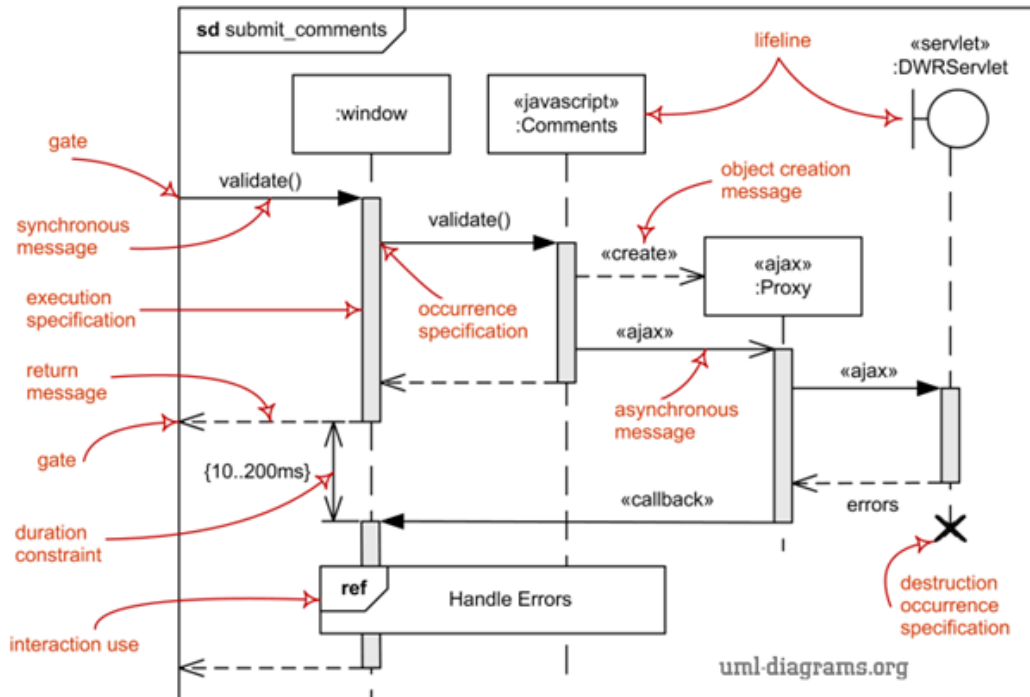


Figure 31. UML/SysML sequence diagram.

² <https://www.ibm.com/docs/en/rsas/7.5.0?topic=diagrams-combined-fragments-in-sequence>

³ <https://www.uml-diagrams.org/sequence-diagrams.html>

UML State Machine Diagrams

In a UML specification of a system [UML], each object is assigned an instance of a state machine that determines the behaviour of the object. An instance of a state machine assigned to the i -th object is denoted by SM_i . To illustrate the elements of a UML state-machine diagram, we assume the classical example of the generalized railway crossing system [Heitmeyer _94] that operates a gate at a railway crossing. Figure 32 shows the state-machine diagram of a train, whereas Figure 33 depicts the state-machine diagram of the gate controller.

A *UML state machine diagram* typically consists of states, regions and transitions connecting source and target states. The set of all states of SM_i is denoted by S_i , whereas $S = \bigcup_{i \in [1,n]} S_i$ is the set of all states from all instances of state machines. One can consider several types of states, namely: simple states (e.g. *Away* in Figure 32), composite states (e.g. *Main* in Figure 33), final states, and initial pseudostates (e.g. *Initial* in Figure 33). Pseudostates are abstractions that encompass different types of transient vertices in the state machine graph, e.g. initial, choice, or history pseudostates. For each object one defines the set of *active* states A_i , where $A_i \subseteq S_i$, $A_i \neq \emptyset$, and $i = 1, \dots, n$. The areas filling the composite states are called *regions*. The regions contained in the same composite state are *orthogonal* (e.g. *Gate* and *Controller* in Figure 33). The regions contain states and transitions, and thus introduce a *hierarchy* of state machines. We assume that a definition of the hierarchy relation is given, and we implicitly refer to this relation by using the terms ancestor and descendant. See the OMG UML2.1.2 specification [UML] for more details.

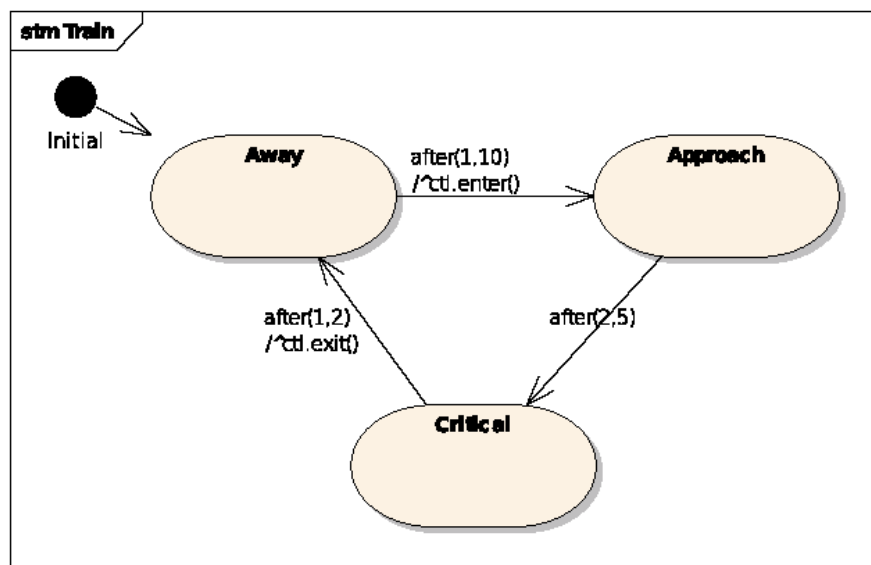


Figure 32. UML State machine diagram of a Train.

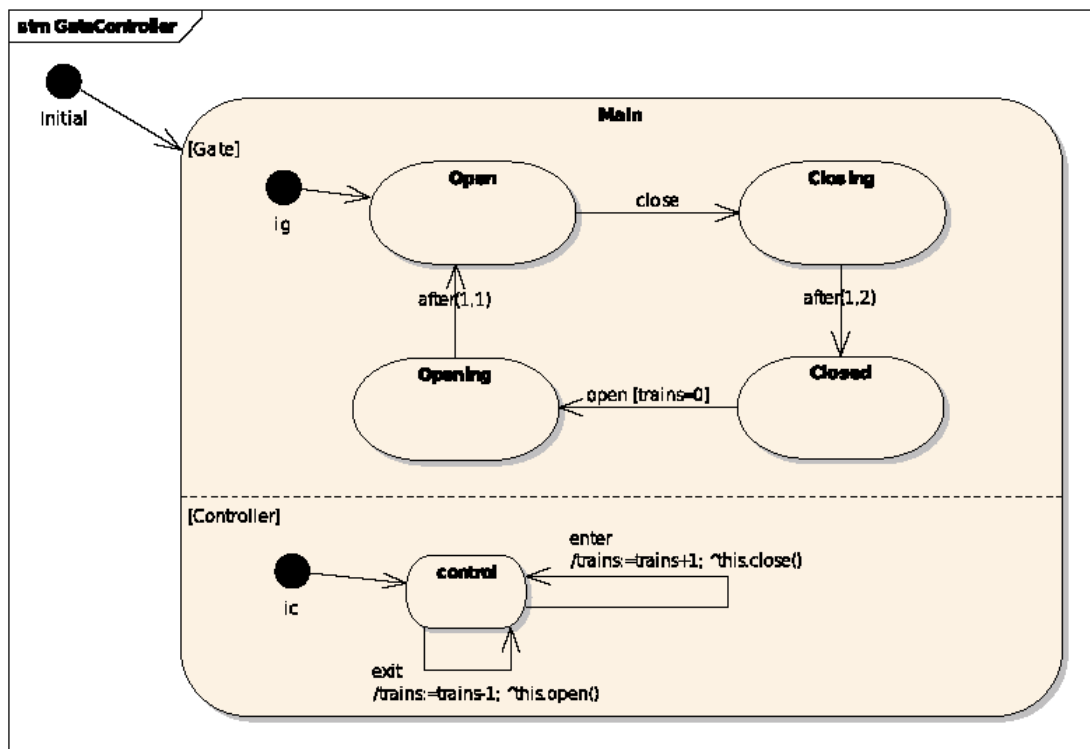


Figure 33. UML State machine diagram of a GateController.

The transitions are labelled with expressions of the form *trigger[guard]/action*, where each of these components can be empty. A transition can be fired if the source state is *active*, the guard (a Boolean expression) is satisfied, and an event matching the trigger occurs. An event can be one of the following three types: an *operation call*, a *completion event*, or a *time event*. In general, firing a transition causes deactivation and activation of some states (depending on the type of the transition and the hierarchy of the state machine). When this happens, the state machine changes its *configuration*.

A time event, defined by an expression of the form *after*(δ_1, δ_2), where $\delta_1, \delta_2 \in \mathbb{N}$ and $\delta_1 \leq \delta_2$, can only occur after passing δ_1 time units have passed and before δ_2 time units have elapsed. This is an extension of the standard *after*(x) expression, which allows one to specify an interval of time in which a transition is enabled. Time is measured by implicit variables called *clocks*. The time flow starts upon from entering a *time state*, namely the source state of a transition with a trigger of the form *after*(δ_1, δ_2). The set of all time states from SM_i is denoted by Γ_i , and the set of all time states from all instances of UML state machines is denoted by Γ , where $\Gamma = \bigcup_{i \in [1,n]} \Gamma_i$.

The operation calls directed to a given object are put into the *event queue* of the object, and then, handled one at a time. If the event at the head of the queue can fire other transitions, these transitions get executed and the event is consumed. If, on the other hand, no such transition can fire, the event is simply discarded. The transitions with nonempty trigger are called *triggered transitions*. We refer to the processing of a single event from the queue or a time event with the term *Run-To-Completion (RTC) step*. An event from the queue can be handled only if the previous one has been fully processed, together with all the completion events that can possibly have occurred as a consequence. A completion event (denoted by k) occurs when a state as

completed all of its internal activities. Completion events fire *completion transitions*, i.e., transitions without an explicit trigger. Completion transitions have priority over triggered transitions. The execution of the whole system follows the classic interleaving semantics [Diethers_02]. During a single step only one object can perform its RTC step. If more than one object can execute such a step, then an object is chosen in a non-deterministic way. However, if none of the objects can perform an *untimed action* (i.e., an action not labelled with a time event), then time flows. Note that this happens when all the event queues are empty and all the completion events have been processed. The time flow causes the occurrence of time events. Time events are then processed in subsequent RTC steps.

UML Activity diagrams

UML Activity diagrams can be seen as an object-oriented version of a flowchart combined with a data-flow diagram that allows to model any process as a sequence of atomic activities connected via different kinds of nodes/edges (starting, ending, split, merging, forking, join, etc.), which define the associated control/data flow. Due to their nature, along with use cases and state machines, activity diagrams can be considered as behaviour diagrams and can be attached to any modelling element in order to describe its specific behaviour. The most common uses for this type of diagrams are:

1. to model software elements, such as methods, functions, and operations;
2. to exemplify the logic of an algorithm or the details of an operation;
3. to give a high-level understanding of the functionalities of a system;
4. to model the flow between different use cases;
5. to illustrate a business process or workflow.

While in UML 1 activity diagram were considered just a special cases of state-machine diagrams, in UML 2 they have an independent and more expressive semantics based on Petri nets. The basic components take, indeed, inspiration from those of Petri nets and can be defined as follow:

- action nodes (usually depicted as ellipses or rounded-corner rectangle): identify atomic units of work within the whole activity, where the underlying system performs a given monolithic task;
- control nodes: describe the flow of control throughout the whole activity; they are partitioned into several classes:
 - start or initial node (filled black circle): symbolizes the beginning of the activity;
 - end or final activity nodes (encircled black circles): represent the final steps in the entire activity;
 - end or final flow nodes (crossed white circles): represent the termination of specific control/data flows;
 - decision nodes (diamonds with one incoming edge and two or more outgoing edges): route the flow among different paths depending on some guard condition;
 - merging nodes (diamonds with two or more incoming edges and one outgoing edge): merge the flow of several paths;
 - fork or split nodes (solid black bars with one incoming edge and two or more outgoing edges): split the flow into multiple concurrent sub activities;

- join nodes (solid black bars with two or more incoming edges and one outgoing edge): synchronizes multiple concurrent flows.

5.3.2 Formal Models

This subsection focuses on the identified formal modelling notations that have the expressive power to support formalizing behaviour specific to systems and variants of the MB and VC operational scenarios.

Petri Nets

Petri Nets (PNs) [Murata_1989] are a well-known and widely used formalism to model complex systems whose behaviour include concurrency, synchronization, conflict, mutual exclusion. They support formal specification and verification of correctness allowing to describe both the static and dynamic characteristic of real systems.

A PN model is a direct bipartite graph whose two types of nodes are called *places* (drawn as circles) and *transitions* (drawn as bars or boxes). With respect to transitions, the *arcs* of the graph can be: *input arcs* (arrows from places to transitions), *output arcs* (arrows from transitions to places) or *inhibitor arcs* (circle-headed arcs from places to transitions). The *multiplicity of an arc* is a natural number $k \geq 1$ associated with the arc. Places can contain *tokens* (drawn as black dots within places). Places represent *local* system states or conditions; transitions describe events or activities that may modify the system state.

The *state of a PN* is distributed and it is defined by the number of tokens in each place, and it is called *marking*. *Enabling and firing rules* are associated with transitions.

An enabling rule stipulates the conditions under which a transition may fire. A transition t is *enabled* if and only if each input place contains a number of tokens greater than or equal to the multiplicity of the arc connecting the place to the transition, and each inhibitor place contains a number of tokens strictly smaller than the multiplicity of the inhibitor arc. Only enabled transitions can fire. A firing rule defines the modification to the marking due to the transition firing, therefore the firing rule defines the changes of state produced by the transition. When a transition t fires, it deletes from each of its input places as many tokens as the multiplicity of the arc connecting that place to t , and adds to each of its output places as many tokens as the multiplicity of the arc connecting t to that place.

Different PNs classes have been defined, which also allow to describe both the logic and temporal evolution of the system, thus enabling forms of performance analysis.

Coloured Petri Nets

Coloured Petri Nets (CPNs) [Jensen_1981, Jensen_2007] are an extension of PNs designed to provide capabilities to create more compact and parametrizable models, making them better suited to model large industrial systems [Vanit_2018]. CPNs combine the process modelling capabilities of ordinary PNs with the power of a high-level programming language called CPN ML, which provides primitives to define data types and manipulation of data values [Gehlot_2010].

In particular, while in ordinary PNs tokens are indistinguishable from one another, CPNs assign to each token a given *value* from a rich set of types (the so-called *colour set* or *token set*), and arc decoration are extended to take into account not only the multiplicity, but also the value of the tokens, and to possibly perform data manipulation operations.

Consider, for example, the CPN in Figure 34 [Gehlot_2010]. In this example, the initial place *ReadyToReact* contains three “H” (hydrogen) tokens and two “O” (oxygen) tokens, which are labelled on the top right of the place. The “++” symbol denotes a multi-set union. The weight on the arc models the fact that two tokens of hydrogen and one token of oxygen are needed for the *Reaction* transition to be enabled and fire. Once the transition fires, the tokens are taken from the *ReadyToReact* and a “W” (water molecule) token is put into the *ResultingCompound* place.

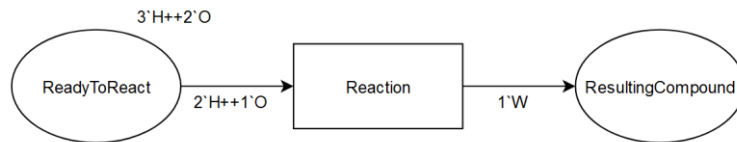


Figure 34. CPN example.

Generalized Stochastic Petri Nets

Generalized Stochastic Petri Nets (GSPNs) [Marsan_1995] are stochastic extensions of Petri Nets, in which two types of transitions are allowed: *timed transitions* are associated with a random, exponentially distributed, firing delay and model time-consuming activities; *immediate transitions* are, instead, associated with a null delay and model the verification of some logical condition or activities requiring a negligible amount of time to be completed. GSPNs with finite state space can be converted into Markov chains and their numerical analysis can be performed to compute performance indices. Alternatively, simulation can be used to estimate performance indices of models with unbounded or very large state space.

The model in Figure 35 shows the structure of a GSPN. This example is taken from [Flammini_2014], and is a (small) part of a more complex model used to analyse the MA delay in ERTMS L2. Its aim is to illustrate the formalism and exemplify the application of GSPN modelling to the railway domain. Specifically, this GSPN is related to the performance of the GSM-R network. The messages, sent by the RBC, are stored in place *GSMR_BUFFER* and delivered after a stochastic delay, associated with the timed transition *COMM*, which is assumed to be exponentially distributed. The firing of transition *COMM* generates a token in place *TX_BUF*, which models the message reception. Upon reception, the on-board subsystem verifies data integrity. If the message is corrupted the immediate transition *TX_FAIL* fires (this event is expected to occur with a given probability *Pfail* associated with the immediate transition). Otherwise, the immediate transition *TX_OK* fires (with probability $1-P_{fail}$): in this case, a token is generated in place *RE_TX* enabling the retransmission request, which is notified to the RBC when transition *R_INJECT* fires. In the model, an inhibitor arc is also present from place *GSMR_BUFFER* to transition *RBC_RX*; therefore, RBC does not add a new message in the buffer until the previous one has been sent to the on-board subsystem.

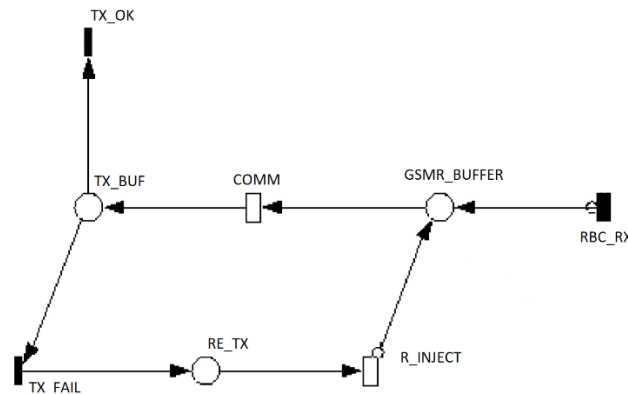


Figure 35. GSPN example.

Stochastic Activity Networks

Stochastic Activity Networks (SANs) are another stochastic extension of Petri Nets, introduced to analyse concurrency, timeliness, fault tolerance, and degradable performance of complex computing systems [Sanders_2001]. A SAN model comprises four primitives which define its structural components: places, activities, input gates, and output gates. Places are the same as in PNs, activities correspond to transitions and can be timed or instantaneous. Gates are introduced in SANs to allow for greater flexibility in defining enabling and completion rules. The expressive power of SANs resides in by the possibility of associating actions (called *cases*) with the activities that may be taken upon the completion of an event, an *output function* with the *output gates*, and an *enabling predicate* and an *input function* with the *input gates*.

The temporal specification of SANs is stochastic, and is defined by associating an activity time distribution function with each timed activity and a probability distribution with each set of cases. Hence, *cases allow to model spatial uncertainty*. These distribution functions introduce uncertainty about which activities will be enabled, when they are associated with instantaneous activities, and uncertainty about the next state entered upon completion, when associated with timed activities. Once activated in a specific marking, timed activities can be restarted, or with a different distribution, according to what specified by the associated *reactivation function*.

The reward structures that relate the possible behaviours of the stochastic process to specified performance variables quantify benefits associated with activity completions and number of tokens contained in the places. This means that the reward structure of a SAN allows for a variety of ways to define different types of performance (reward) variables. Unlike GSPN, a SAN model can only be analysed by simulation.

Figure 36 shows the structural elements of a SAN. Places are drawn as circles, instantaneous transitions are drawn as solid bars, timed transitions are represented by thick bars, input and output gates are depicted as triangles. Cases are denoted by as small circles on one side of the associated activity.

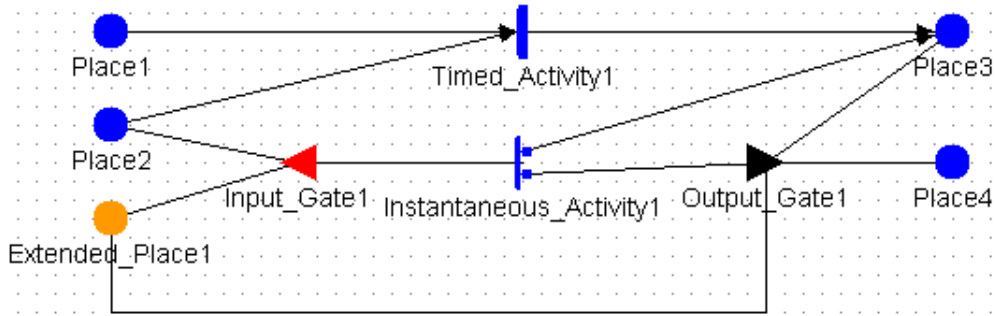


Figure 36. SAN example.

Extended places (e.g., *Extended_Place1* in Figure 36) differ from standard places because the tokens they may contain represent atomic variables, data structures or arrays of primitive data types (i.e., short, int, long, float, double, bool and char). The firing of timed activities is associated with general distributed random variables (e.g., Exponential, Normal, Binomial) whose parameters can be numeric constant or dependent on marking. The probability associated with each case (e.g., the cases associated with *Instantaneous_Activity1*) could be specified as a numerical constant or a function, as well. If no cases are present, the default is assumed with a probability equal to one. Input and output gates can be used to control the enabling condition of an activity and to change the state of the system when the activity fires. An activity is enabled when the predicates of all input gates connected to the activity are evaluated to true, and each place connected to the incoming arcs contains at least one token. When an activity fires, the input and the output functions of the input and output gates (respectively) are executed, while tokens of connected places are updated as in the Petri Net firings. In Figure 36 the timed activity *Timed Activity1* is enabled by tokens in places *Place1* and *Place2*. When the activity fires, a new token is added in *Place3*. At the bottom of the figure, the instantaneous activity *Instantaneous_Activity1* is enabled by the predicate of the *Input Gate1*, which, in turn, is evaluated with respect to the marking of *Place2* and *Extended_Place1*. When the activity fires, two cases are possible. If selected, the first case adds a token to *Place3*; alternatively, the second case enables the execution of the output gate *Output Gate1*, which, in turn, updates the marking of *Place4* and *Extended Place1* according to the output function associated with the activity.

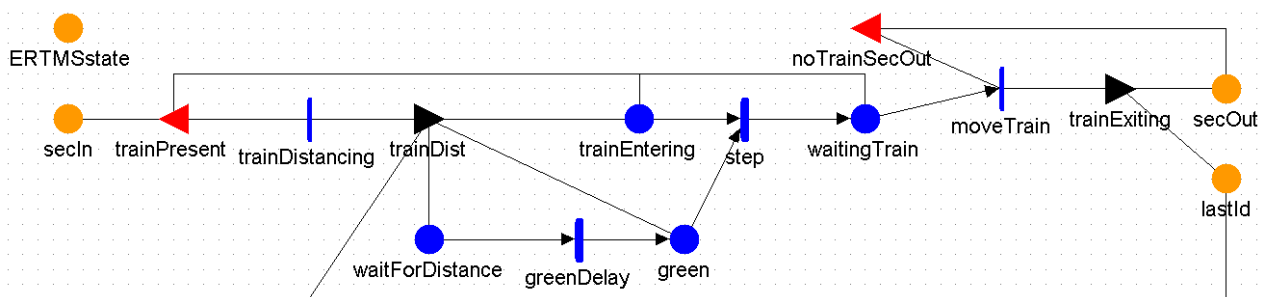


Figure 37. Straight railway line.

The SAN in Figure 37 models a straight ERTMS L3 railway line with an incoming and an outgoing connection (extended places *secIn* and *secOut*). The model is taken from [Flammini_21], and its aim here is just to show an application of SAN to the railway domain.

GSPNs and SANs are well supported by automated tools that provide both modelling and

analysis/simulation features. GreatSPN⁴ or Pipe⁵ may be used for GSPN, but many others tools are available, also for different classes of timed Petri Nets. A well-known tool used to edit and analyse SAN models is Möbius⁶, which also supports the compositional and hierarchical development of models, therefore SANs are well suited to compositional modelling. In Möbius all enabling predicates, input and output functions, parameters, types, and variables are expressed by C++ statements, thus allowing the introduction of actual code in the model definition.

Timed Automata

Timed automata (TA) are a well-known formalism to model and verify safety critical systems with timing constraints [Alur_1994]. TA extend finite state automata with *clocks* (i.e., real-valued variables, all of which evolve linearly at the same rate).

The behaviour of real-time systems is modelled by finite graphs augmented with a finite set of clocks. The vertices of the graph are called *locations* and represent the possible control modes of the system, while edges are called (control) switches and model discrete changes of control modes. Time can only flow within locations, while switches are instantaneous. Clocks can be compared with rational constants to form *clock constraints*. These constraints are expressed as conjunction of linear inequalities: for a set X of clocks, the set $\Phi(X)$ of clock constraints φ is defined by the grammar

$$\varphi := x \leq c \mid c \geq x \mid x < c \mid c < x \mid \varphi_1 \wedge \varphi_2$$

where x is a clock in X and c is a rational constant in \mathbb{Q} .

Clock constraints can be used to express enabling conditions for switches, called *guards*, and to specify location *invariants*, namely bounds on the time the system can spend in a given location.

Formally, a timed automaton A is a tuple $\langle L, L_0, \Sigma, X, I, E \rangle$, where:

- L is a finite set of locations;
- $L_0 \subseteq L$ is a set of initial locations;
- Σ is a finite set of input symbols;
- X is a finite set of clocks,
- I is the invariant mapping, associating each location s with a clock constraint in $\Phi(X)$;
- $E \subseteq L \times \Sigma \times 2^X \times \Phi(X) \times L$ is a set of switches. A switch $\langle s, a, \varphi, \lambda, s' \rangle$ represents an edge from location s to location s' that can be taken on reading the input symbol a . φ is a clock constraint over X that specifies when the switch is enabled, and the set $\lambda \subseteq X$ gives the clocks that must be reset, i.e. set to 0, when the switch is taken.

The example in Figure 38 is taken from [Alur_1999]. It models an automated level crossing, a case study widely used as a benchmark for safety analysis approaches (including Petri Nets).

⁴ <http://www.di.unito.it/~greatspn/index.html>

⁵ <http://pipe2.sourceforge.net/>

⁶ <https://www.mobius.illinois.edu/>

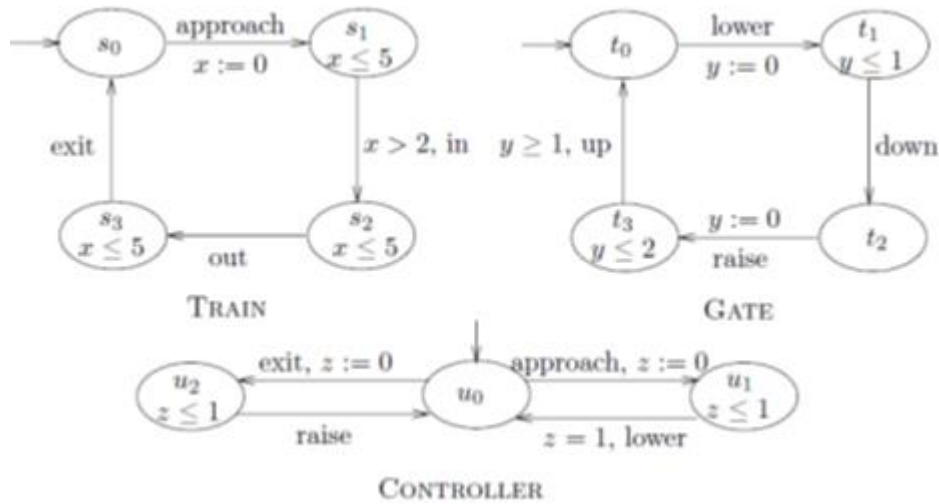


Figure 38. TA example.

The system consists of three components: Train, Gate (barrier) and Controller. The safety requirement is that the gate must be closed whenever the train is inside the gate. Therefore, in every reachable state, if the location of Train is s_2 then the location of Gate should be t_2 . This location is indeed reachable in the product graph. The reachability problem is to determine whether or not some target location is reachable. Verification of safety requirements of real-time systems can be formulated as reachability problems for timed automata.

For instance, given the timing constraints in the automata in the figure, the event *approach* cannot be immediately followed by the event *in*. In particular, both clocks x and z have the same value, when the automaton Train is in location s_1 and Gate is in location t_0 . Therefore, the event *lower* with guard $z = 1$ is guaranteed to precede the event *in* with guard $x > 2$.

Several state-of-the-art model checkers support the analysis of TA. For example, UPPAAL⁷ is an integrated tool environment for modelling, validation and verification of real-time systems modelled as *networks of timed automata*, namely parallel compositions of TAs, extended with data types (bounded integers, arrays, etc.) and message passing primitives to model communication and synchronization among the modelled components.

Stochastic Priced Timed Automata

A stochastic priced timed automaton (SPTA) is defined as the following tuple:

$$\text{SPTA} = \langle L, l_0, X, \Sigma, E, R, I, \mu, \gamma \rangle$$

where L is a finite set of locations, $l_0 \in L$ is the initial location, X is a finite set of continuous variables, $\Sigma = \Sigma_i \cup \Sigma_o$ is a finite set of actions partitioned into inputs (Σ_i) and outputs (Σ_o), E is a finite set of edges of the form (l, g, a, ϕ, l') , where l and l' are locations, g is a predicate on R^X , action label $a \in \Sigma$, and ϕ is a binary relation on R^X , $R : L \rightarrow N^X$ that assigns a rate vector to each location, I assigns an invariant predicate $I(l)$ to any location l , μ is the set of all density delay functions $\mu_s \in L \times R^X$, which can be either uniform or exponential distribution, and γ is the set of

⁷ <https://uppaal.org>

all output probability functions γ_s over the Σ_o output edges of the automaton.

The semantics of SPTA is defined over a timed transition system with a stochastic interpretation based on: (i) probabilistic choices between multiple enabled discrete transitions, and (ii) nondeterministic time delays that can be refined based on probability distributions, either uniform distributions for time-bounded delays or (user-defined) exponential distributions for unbounded delays.

Assuming that inputs are enabled, clock sets and output actions are disjoint, a collection of composable SPTA can be defined as a network of SPTA (NSPTA) ($A_1 \parallel A_2 \parallel \dots \parallel A_n$). The states of the NSPTA are defined as a tuple $s = \langle s_1, \dots, s_n \rangle$, where s_j is a state of A_j of the form (l, v) , where $l \in L_j$ and $v \in \mathbb{R}^{x_j}$, where different automata synchronize based on standard broadcast channels. The probabilistic semantics is based on the principle of independence between components. Each component decides on its own (based on a given delay density function and the output probability function) how much to delay before producing an output. Figure 39 shows the level-crossing (train-gate controller) example with an underlying stochastic behaviour.

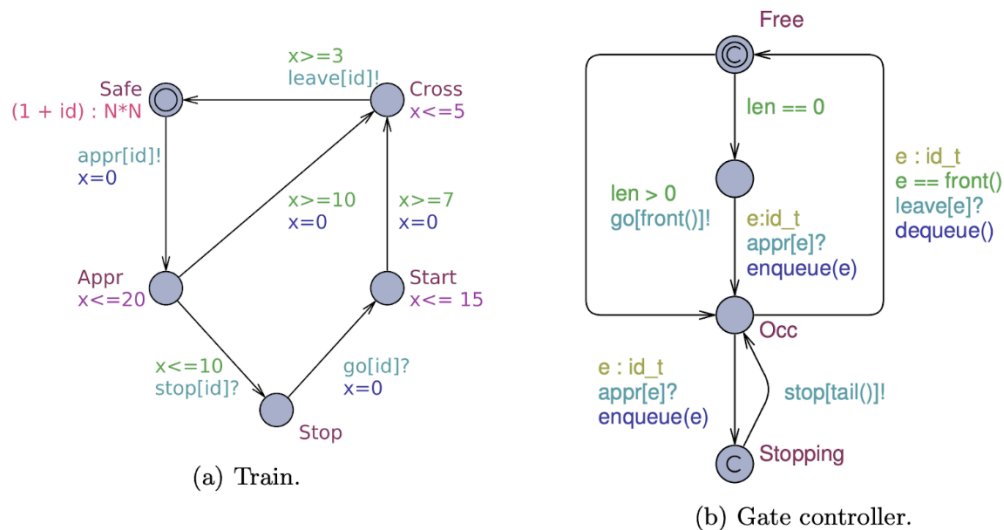


Figure 39. Level-crossing components as SPTA.

Hybrid Automata

A hybrid automaton is a formal model for a mixed discrete-continuous system. A hybrid automaton H comprises the following components [Henzicher_1996]:

- A finite set $X = \{x_1, \dots, x_n\}$ of real-valued variables. The number n is called the dimension of H . The set $X' = \{x'_1, \dots, x'_n\}$ represents first derivatives of the real-valued variables describing their continuous behaviour over time.
- A finite directed multi-graph (V, E) , whose vertices in V are called control modes and whose edges in E are called control switches.
- Three vertex labelling functions $init$, inv , and $flow$ assigned to each control mode. Each initial condition $init(v)$ is a predicate, whose free variables are taken from X . Each invariant condition $inv(v)$ is a predicate whose free variables are again taken from X . Each

flow condition $flow(v)$ is a predicate over the variables in X and their derivatives in X' that describes the continuous evolution of the real-valued variables.

- An edge labelling function $jump$ that assigns to each control switch a predicate.
- A finite set L of events, and an edge labelling function $event: E \rightarrow L$ that assigns to each control switch an event.

The Figure 40 shows an example of a hybrid automaton for level crossing barriers. The automaton illustrates the different control modes of the automatic level crossing barriers where the real-valued variable θ denotes the angle formed by the barrier and the horizontal axis. Control modes are *opening*, *open*, *closing* and *closed*. Jump functions are θ equal to 85 and θ equal to 0. Events are *approach* and *leave*. Flow condition, expressed in terms of constraints on the first derivative θ' of the variable θ , specifies is the speed of barrier opening/closing (± 5).

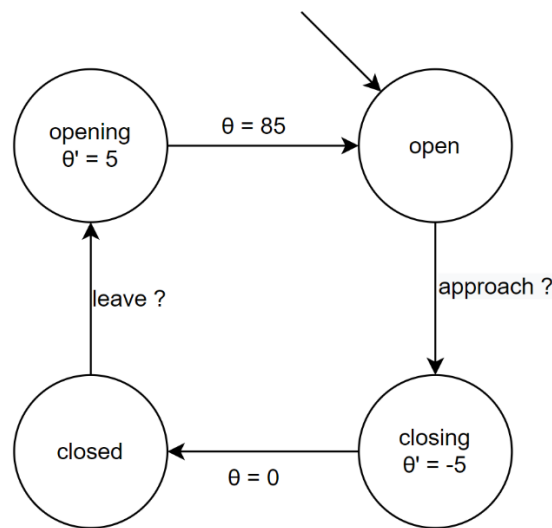


Figure 40. Hybrid automaton for barrier opening.

Assuming a probabilistic interpretation of delays and discrete transitions of hybrid automata similar to that of SPTA presented previously, both types of automata can then be formally analysed by employing UPPAAL SMC (Statistical Model Checker) [David_12], which is an extension of UPPAAL model checker. A probabilistic extension of weighted metric temporal logic (WMTL) [Bulychev_12] is used to encode the queries as follows:

- *Hypothesis testing*: check if the probability to reach a state ϕ within cost $x \leq C$ is greater or equal to a certain threshold p ($\Pr(\star_{x \leq C} \phi) \geq p$),
- *Probability evaluation*: calculate the probability $\Pr(\star_{x \leq C} \phi)$ for some NSPTA,
- *Probability comparison*: is $P(\star_{x \leq C} \phi_1) > P(\star_{y \leq D} \phi_2)$?

where \star stands for either future ($\langle \rangle$) or globally (\square) temporal operator.

Promela

Promela (Process Meta Language) is a formal specification language for distributed systems based on Dijkstra's guarded command language. Promela provides communication and concurrency primitives inspired by process algebras, and it is tailored to *asynchronous composition* of processes.

The language allows for dynamic creation of processes that can communicate by means of

shared variables and message passing using queued or rendezvous channels. Models consist of processes, message channels, and variables. Processes specify behaviour, channels and global variables define the environment in which the processes run. Variable types include predefined data types (int, bit, byte, bool) and array. In addition, user-defined data types are supported (enumerations and record structures).

Promela specifications can be verified using the model checker SPIN⁸, an on-the-fly verifier developed at Bell-Labs. The verifier can be used to prove the correctness of system invariants and supports the verification of linear time temporal constraints, either expressed as *Promela never-claims* or directly formulated in temporal logic. A Linear Temporal Logic (LTL) formula is translated into a Buchi Automaton, which is turned into a Promela process (a never-claim) and synchronously composed with the rest of the system. Promela has a C-like syntax that makes its use suitable for compositional and parametric modelling.

The process reported in Figure 41 is part of a Promela specification modelling the initiation of the communication session between a train and RBC. The process in the Figure 41 is in charge of handling the dynamic activation of the process that models the session establishment (command *run session_establishment()* in the example) and the subsequent activation of the Start-of-Mission procedure, or the abortion of the procedure in case the of the communication session cannot be successfully established.

```
active proctype manage_som()
{
  do
    :: ( state_MANAGE_SOM == Initial_MS && next_MANAGE_SOM==1 ) ->
    atomic {
      idcount=idcount+1;
      printf("<vv:steps name='Initial_MS' type='Initial_MS' id='%d'>/>\n",idcount);
      next_MANAGE_SOM=0;
      printf("<vv:steps name='invalid' type='transition'>/>\n");
      run session_establishment();
      transition_MANAGE_SOM = T01_MS;
      state_MANAGE_SOM = establishment;
    }
    :: ( state_MANAGE_SOM == Final_MS && next_MANAGE_SOM==1 ) ->
    atomic {
      idcount=idcount+1;
      printf("<vv:steps name='Final_MS' type='Final_MS' id='%d'>/>\n",idcount);
      next_MANAGE_SOM=0;
    }
  break;
od unless { AbortChan_MANAGE_SOM?[1];printf("Machine='MANAGE_SOM' aborted");}
}
```

Figure 41. Promela example.

The example was taken from the ARTEMIS project Crystal, where the goal of this modelling activity was the *automated generation of test cases* for validation purposes. Indeed, model checking can also be used to pursue other objectives, besides property verification. In particular, the capability of a model checker to provide a counterexample for a violated property can be exploited to generate an execution trace that can be interpreted as a test sequence.

Event-B

⁸ <http://spinroot.com/spin/whatispin.html>

Event-B [Abrial_2010] is a formal method used for the stepwise development of models. It combines concepts coming from the Action Systems [Back_1983] and the B-Method [Abrial_1996]. As for tool support, the Rodin platform [Abrial_2010] provides for automated modelling and verification capabilities. Event-B modelling starts with an abstract model, which is iteratively refined with more details until the system specification is completed. Event-B models consist of two parts: contexts and machines. Contexts store the static part of the model and machines model its dynamic. In the context part of the model we define carrier sets, axioms, constants and theorems. Machines specify the behavioural properties of the system, by means of variables, invariants and events. Machines can be refined and may have visibility of one or more contexts and contexts can be extended as well. In this way a user can start with an abstract model of a system and, then, add more details subsequently. A general representation of an Event-B machine and context is shown in Figure 42.

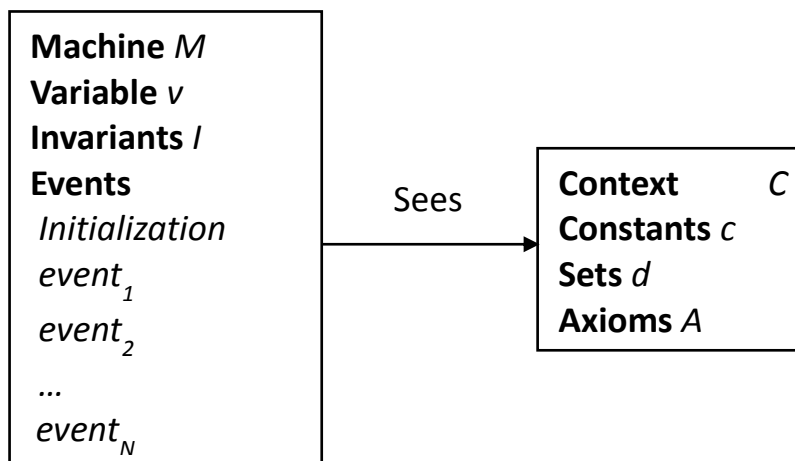


Figure 42. Abstract Structure of an Event-B Model

A key feature of Event-B is its support for formal refinement. In Event-B, the refinement is a procedure of transforming an abstract model into a more comprehensive model via stepwise transformations, while preserving the correctness properties of the original abstract model.

Event-B has as tool support the Rodin platform [Abrial_2010]. The Rodin platform is based on Eclipse and it allows for editing, proving properties (by generating proof obligations), and animating the model, as well as perform model-checking tasks. Several proof engines can be used to automatically prove different properties of the model, by discharging the generated proof obligations. If some proof obligations cannot be discharged automatically, then those can be discharged interactively by the modeller. The fact that some proof obligations are not discharged automatically signals that there might be some problem with some modelling aspect of the system. The modeller then has a chance to edit the model to address the issue. This interleaving between modelling and proving is an important aspect of the Rodin platform and is somewhat similar to the coding-compilation process in computer programming [Abrial_2010].

Event-B has numerous applications in many fields. For example, refinement in Event-B was used to model file transfer and bounded retransmission protocols, control systems, concurrency, electronic circuits, network synchronization etc.

To understand Event-B, a simple case study of a train control system is discussed below. This case study is presented in detail in [Simon_2011]. In Figure 43, a control system is shown. The major requirements for the management of this control system are the following: the station has a single one-way track, an approach block, a station block and an exit block as shown in the figure. The train approaches the station from the approach block, enters the station via the station block and exits through the exiting block. There are also two signals present at the two ends of the station. The signals close (turn to red) automatically when the train passes by, while the opening of signals (turn to green) is controlled by the system. A train occupies no more than one block at any given time and the track is one-way only. In addition, a train at the approach block can only enter the station block if the *in* switch is set to that particular block. Two trains cannot normally occupy the same block at the same time. Train detection sensors report the vacancy status of the block, which is either occupied or vacant. The trains are assumed to stop at red signals. The controller is equipped with actuators to change the signals' status. All of these requirements are specified in the form of invariants in the Event-B model.

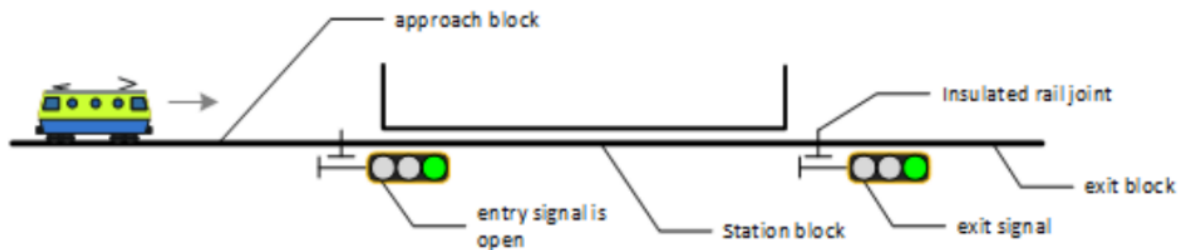


Figure 43. Control system example.

The control system along with a model of the environment are developed in four different stages. The communication between the environment and the control system is two ways, and various sensors and actuators are used in this communication as illustrated in Figure 44.

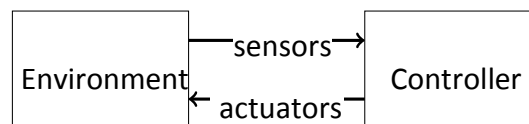


Figure 44. Communications.

The model of the environment that consists of different Blocks, Switches, Signals and Trains, is built in the first stage. For example, for the first sub-stage of Blocks, the set of blocks (Blocks) consists of the approach block (APP), the exit block (EXT), and the station block (STNS), while the set of occupied blocks is recorded using the variable OCC. The status of a block is modelled by means of four different events: ARRIVE, MOVE_IN, MOVE_OUT, and LEAVE. At the occurrence of the ARRIVE event, block APP becomes occupied as the train approaches. When the MOVE_IN event occurs, block APP becomes available again and the station block becomes occupied as the train moves to into the station. The MOVE_OUT event makes block EXT occupied and station

block available, because of the train exiting the station. Finally, the LEAVE event makes block EXT available, because the train leaves the area. In the Switches stage, the two switches located at the two ends of station are modelled. In the sub-stage Signals, the mode of the signal, which can be either Red or Green, is modelled, while in the Trains stage, the MOVE_IN and ARRIVE events are refined in such a way that the train's passage becomes safe. Two of the events of this stage are shown in Figure 45.

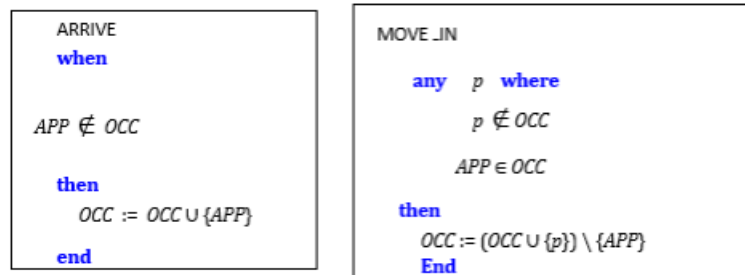


Figure 45. The ARRIVE and MOVE_IN Events.

The second stage of the model takes care of the Actuators. In this stage of the model some actuators are introduced in order to capture the changes of the state of the environment component enforced by the control system. This is done in such a way that the normal behaviour of the environment is forced into the modelled behaviour. In the third stage the Sensors and the Controller are introduced. In this stage, different variables are introduced for the sensors of the blocks, entry and exit signals. The events of the controller stage are responsible to maintain the status of different points, and then to send commands to the signals. At the end of this stage, the model of the control system is complete, along with a working environment that is crucial to ensure the safety requirements. In the last stage, Scheduling, the guards of the controller are strengthened in order to optimize its execution.

5.4 Hazard Modelling

A hazard⁹ analysis for safety-critical systems aims to investigate all internal and external system factors leading to accidents¹⁰. In such analysis, hazards are first identified using different techniques. Once identified, their associated risk¹¹ should be prioritized so that risk mitigation can efficiently attain the desired level of safety. This involves hazard elimination, hazard occurrence reduction, hazard control, and damage reduction. If a hazard cannot be completely eliminated, its occurrence shall be prevented or minimized by specifying hazard reduction measures to prevent or minimize the conditions that could lead to the hazard.

⁹ Hazard (or hazardous situation) is a state or set of conditions of a system that, together with other conditions in the environment of the system, which will lead to an accident.

¹⁰ Accident is an undesired and unplanned event that results in a specified level of loss (death, system loss, injury, damages caused to the environment).

¹¹ Risk is the frequency of occurrence of accidents and incidents resulting in harm (caused by a hazard) and the degree of severity of that harm.

The following table presents different hazard analyses and techniques for safety-critical systems [Ravikumar_2016] [Leveson_2004]. “Forward and Backward Searches” and “Top-Down and Bottom-up Searches” are techniques that are used in some analyses. The second table below presents the advantages and drawbacks of the different analyses.

Hazard analysis and technique	Key elements	Main steps
Forward and Backward Searches	<ul style="list-style-type: none"> - Temporarily ordered events 	<ul style="list-style-type: none"> - The safety application has to be associated to a temporal structure. - Tracing a particular event in forward time for determining reachable states from an initial state
Top-Down and Bottom-up Searches	<ul style="list-style-type: none"> - Basic events, sets, tasks, or systems - Relation of each individual component failures with the overall behaviour of the system 	<ul style="list-style-type: none"> - The top-down approach allows breaking down an event, set, task, or system into more basic events, conditions, tasks, or sub systems. - The bottom-up approach is a forward search.
PHA (Process/ Preliminary Hazard Analysis)	<ul style="list-style-type: none"> - Hazard category/level (e.g. acceptable, tolerable, unacceptable) using their likelihood (e.g. frequent, probable, occasional, remote, improbable) and severity (e.g. catastrophic, critical, marginal, negligible) - Prioritization of hazards according to their category/level 	<ul style="list-style-type: none"> - Identifying hazards that might exist during system operation - Defining specification and criteria to be followed during the system design - Identifying management and technical responsibilities for hazard control actions - Identifying control measures
FTA (Fault tree Analysis)	<ul style="list-style-type: none"> - Logical structure of the tree defined according to the failure events in the system - Probability of occurrence of basic failure events - Feared event on top of the tree and its probability 	<ul style="list-style-type: none"> - Top-down analysis of the hazard causes with a tree starting from a top feared event divided into more basic events using logical gates (AND and OR gates are the most frequent) - Qualitative analysis: tree reduction to identify minimal cut-sets and weaknesses in the system - Quantitative analysis: calculation of the probability of occurrence of the top event
ETA (Event Tree Analysis)	<ul style="list-style-type: none"> - Alternative paths with success or failure of protection systems - Probability of occurrence of the failure event in each path - Initiating events (e.g. system failures, external events) - Probability of risky final events 	<ul style="list-style-type: none"> - Forward search to identify the various possible outcomes of a given initiating event
CCA (Cause-Consequence Analysis)	<ul style="list-style-type: none"> - Critical events - Causes and effects of critical events - Logical relationship between events 	<ul style="list-style-type: none"> - This analysis starts with a critical event and determines the cause of the event by using top-down or backward search approach. - Then it shows both the time

		dependency and casual relationship among events (AND and OR gates to describe the cause/event relations, and vertices to describe the event/consequence relations)
HAZOP (Hazards and Operability Analysis)	<ul style="list-style-type: none"> - Systems failures and more complex types of hazardous events with their causes and consequences 	<p>This analysis is based on a systems theory model of accidents that assumes accidents are caused by deviations from the design or operating intension of safety critical systems, its mains steps are:</p> <ul style="list-style-type: none"> - Identifying all possible deviation from the design expected operation and all hazards associated with the deviations - Determining causes and consequences of deviations - Risk ranking applied on the severity of the causes - Defining appropriate actions to mitigate and manage the risk
FMECA / FMEA (Failure Mode Effect and - Criticality- Analysis)	<ul style="list-style-type: none"> - Categories in the FMECA / FMEA table 	<ul style="list-style-type: none"> - Identifying and listing all components and failure models with their possible operating modes - The Likelihood and severity of events are estimated and lead to the event criticality (if analysed, otherwise it is a FMEA) - The results are documented in a table with column heading such as component, failure probability of the component, failure operating mode, effects
SMHA (State Machine Hazard Analysis)	<ul style="list-style-type: none"> - Set of states - Transition between states 	State Machine Hazard Analysis involves forward search that starts from the initial state of the system, generates all possible paths from the states and determines whether any of the state is hazardous. Its algorithm can be implemented using Petri-net models.
STPA (System-Theoretic Process Analysis)	<ul style="list-style-type: none"> - System control structure linked to accident, based on STAMP hazard model (System-Theoretic Accident Models and Process) 	<ul style="list-style-type: none"> - Identifying the losses that the analysis aims to prevent - Building the control structure model of the system - Analysing the control actions in the control structure to examine how they could lead to the losses (UCA-unsafe control actions) - UCA are used to create functional requirements and constraints for the system - Scenario identification

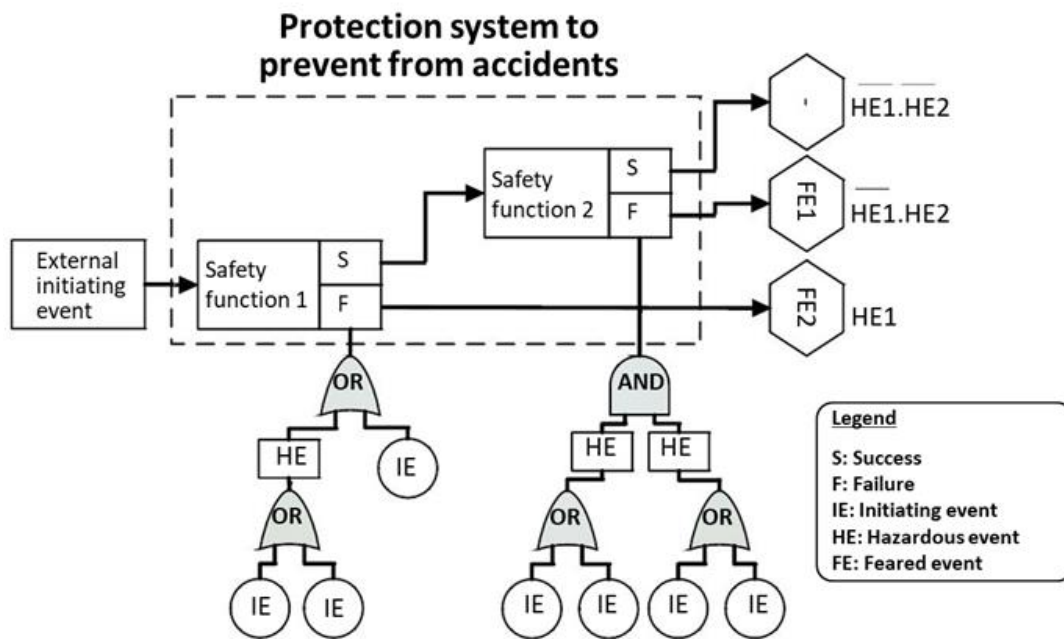


Figure 46. Example of cause-consequence diagram.

	PHA (Process/ Preliminary Hazard Analysis)	FTA (Fault tree Analysis)	ETA (Event Tree Analysis)	CCA (Cause- Consequence Analysis)	HAZOP (Hazards and Operability Analysis)	FMECA / FMEA (Failure Mode Effect and -Criticality- Analysis)	SMHA (State Machine Hazard Analysis)	STPA (System-Theoretic Process Analysis)
Type of analysis	Qualitative/ Semi-quantitative	Qualitative/ quantitative	Qualitative/ quantitative	Qualitative/ quantitative	Qualitative	Qualitative/ Semi- quantitative	Qualitative/ quantitative	Qualitative
Advantages	Systematic analysis	The causal chain of failures can easily be visualized	The chain of events can easily be visualized	Help to identify scenarios because sequence of events are visible	Early identification of design problems	Structured safety requirements	determine safety requirements directly from the system design	It allows analysing several factors: safety issue related to design error, software flaws, component interaction and human decision making errors
Drawbacks	It lies heavily on the judgement of the engineers performing the analysis, hazard causes can be omitted	Binary method that makes it difficult to manage multi-state components and their dependencies	Only one initiating event analysed at a time and difficulties to manage dependencies	Difficult to be used for complex systems	It lies heavily on the judgement of the engineers performing the analysis	Tedious and costly if applied to all parts of a complex design	Impractical for large complex systems	It is difficult to identify basic risk for new components and defining control structure
Automation	No	Yes	Yes	No	No	Yes	Yes	No
Languages or artefacts used		Logical structure obtained manually or from Reliability Diagrams, Binary Decision Diagrams for quantification	Branching structure	Logical structure and branching structure	Functional and operational specifications		Labelled transition systems	

6. Guidelines for Data Modelling

EULYNX has collected substantial know-how in terms of modelling static and dynamic railway signalling data. Static configuration data are modelled in EULYNX Data Prep. Dynamic data exchange on the EULYNX interfaces are captured by the EULYNX SCI interface description and SysML state machines.

The purpose of EULYNX Data Prep (henceforth DP) is to standardise exchange of static configuration data between Infrastructure Manager (IM) and signalling supply industry. The UML model (see dataprep.eulynx.eu) applies a set of rules designed to allow transformation into a Platform Specific Model such as XSD or OO-code. Whilst the use case of PERFORMINGRAIL differs from the EULYNX use case, it is sensible to apply below rules taken from EULYNX UML because they produce robust models that are easily ingested into subsequent dynamic models.

Relation to ontologies. A structural model stores domain knowledge in classes and their relations. By this virtue, a structural model is very close to an ontology. Each class must have a clear semantic. A relation between classes should be close to natural language grammar: subject-predicate-object. When modelled so, it is relatively easy to share information by means of semantic triples.

The consequence of “thinking ontology” when writing a structural model is that classes and relations should be named such that they form simple phrases such as “a route has an entry signal”. Route and signal are classes and “has an entry signal” is a named relation between the two.

6.1 Modelling attributes and relations

Attribute should be used when modelling:

- A property of a class that is
- a simple atomic datatype such as string or int and that is
- unlikely to evolve or need further specialisation

E.g. a thing has a name which always is a simple ascii string. Name can be an attribute of thing. Attribute atomic datatypes should exist as such in common languages such as Java, XSD, Python. Attributes that are of complex datatype may be hard to handle by subsequent transformations.

Composite aggregation, a black diamond, should be used when modelling:

- A property of a class that is
- not shared with other classes and that is
- a complex datatype because the transformers can only handle simple datatypes, and/or
- may evolve to accommodate new insights

E.g. a property of a route such as speed could naively be modelled as an integer attribute expressing speed in km/h. However, some IM's prefer other units such as *mph*. This suggests that the route have a composite aggregation with a class Speed that has an attribute unit. And if the need arises to model things such the speed's standard deviation, the Speed class can be further

specialised. Another rule of thumb is to recall that "if the owning object disappears, so will the owned property".

Use **Shared aggregation**, a hollow diamond, when modelling

- A property of a class that is an instance of another class (Subject - predicate - object in terms of ontological modelling) and
- that property is likely to be *shared* with other classes.
- the part (i.e. the target object) remains when the source disappears.

This type of aggregation is rarely used in common modelling and must not be used in EULYNX at all.

Association should be used to express a loose relation between source and target. Hence, association should be *preferred to shared aggregation*. The choice between shared aggregation and implementation in terms of OO code or XML has no impact, so there's little point in spending much time on debating the fine differences. This said, operations applied to the aggregate such as destruction apply to the parts (target). So, when the UML is translated into OO code, this would imply that destructors would delete both composite and parts, which is likely to be undesirable.

Identifiers, quantities and units. Objects should carry an identifier.

- Simple attributes, i.e. attributes of a primitive data type (int, string, ...), need no ID.
- Compositions, that are equivalent to attributes, may need an ID.

This implies that compositions might inherit an identifier from a BaseObject.

Attributes and compositions are owned by a class. *As such*, they don't need an ID, because they can easily be found by navigating to the owning class and then inspecting its children. Such auxiliary classes don't merit an ID. This would apply to simple classes that are target of compositions *only*. Such classes are a kind of attributes.

However, there can be a compelling case for identifying compositions when it is useful to have a handle that one can independently refer to.

For instance, in automatic quality checks, one can then directly refer to datum with parameter ID=7ffe25b6-f2dc-48dd-994a-38abd6776850. This is more fool-proof than having to refer to parameter as a child of the owning class.

Exceptions

*Quantities and units don't have an ID. In an expression like **TPS extends over a length of 20 metres**, the length 20 metres has no ID. This is to prevent bloating the data with UUIDs that provide little benefit. The danger of miscommunication is small because the used QUDV classes such as length and gradient have no children. Furthermore, this is closer to best practice, e.g. when using the Geographic Markup Language (GML), coordinates are owned by their parent-objects but the coordinate-tuples have no identity.*

This also implies that QUDV classes can only be instantiated as children of other classes, i.e. target of compositions; it's impossible to refer to instances of a distance (or gradients, duration, speed).

Presently, spot locations in RTM have identities which would lead to numerous UUID. We're in the process of removing this behaviour.

Rule of thumb

- Classes that are target of associations always inherit from BaseObject
- Classes that are relatively simple and target of compositions only, need not inherit from BaseObject.
- Classes that reflect technical subject matter and facts and/or need to be referenceable (e.g. during quality checks) need an *anchor* that can be referred to. By this virtue, they inherit from BaseObject.

Containers and life cycle. (Almost) each class modelled needs an owner. Ownership is modelled by an aggregation point from the owner to the owned part. Modelling ownership or putting objects into boxes that “own” the objects implies that the owned objects are destroyed when the owner is deleted. Keep the life cycle in mind when doing structural modelling.

Naming conventions. Class diagrams are about a *concern*, subject or facet. Drawings in the diagram inform about definitions and relations.

Rule of thumb is that the diagram should fit on a A3 sheet (landscape) and a beamer.

Diagram names must match the title, which of course must cover the concern.

Enumerations vs specialisation. Enumerations can cause problems with

- ownership, e.g. the asset managers may well define a list of equipment types that only partly overlaps the EULYNX list
- maintenance, e.g. a new equipment type requires an update of the common model.

This is the rationale for below guidance.

Rules of thumb

- Use enumerations if the available choices must be restricted to a set of values.
- Consider a string or numerical attribute when no such restriction is needed.
- Consider whether specialisation into a number of subclasses is needed for future flexibility. For instance, the concept of parenthood is probably better modelled as “a son is a special kind of child and a daughter is a very special kind of child” than “child has attribute gender[m|f]”
- Enumerations can be split up. Class building has an attribute isOfType that is an enumeration with entries [home, office, factory, other]. Now class building can be specialised into house and get an attribute isOfHouseType [mansion, cottage, FeWo]. The end user can create a house that is of type other and of house type cottage.

Grasp. We apply GRASP patterns for robust modelling. Notably

- Information Expert: the class needs to know the information to fulfil the associated requirements. By this rule, many EULYNX classes have no (outgoing) associations with classes that they *need not know*. For instance, a signal does not need to know how it’s configured but a configuration *knows* the signal.
- Low coupling: avoid mutual dependency and optimise reuse. By this principle, we prefer associations or aggregations to compositions. See for instance the low coupling between EULYNX and RSM.
- Polymorphism: who’s responsible when behaviour changes by type. This is closely related to the principle of *subsidiarity*: EULYNX has a common namespace for classes that are common to all. These can be specialised in national domain namespaces. EULYNX

common provides for instance an abstract *Aspect* class that is specialised at national level because every IM has signals but they look different in every country.

Dynamic data guidelines. Object status has, by definition, a time dimension. Systems exchange “snapshots” of object status. Objects are commonly modelled by dynamic state vectors that can be communicated between parties. E.g. a point can have position left/right/unknown. The attribute radius (given as a tangent e.g. 1:34), whilst relevant to operations, isn't considered part of the status because it's invariant.

Some rules of thumb apply to the time aspect of the dynamic data model. This is best explained by an example. A train receives at a given moment a Movement Authority (MA). The MA informs up to where, with what speed the train can safely travel and what information points, i.e., Eurobalises, it will encounter *en route*. Communication towards a travelling train is by nature non-vital and may be interrupted at any moment. This is the defining constraint when designing dynamic objects that dictate dynamic information: when not detected, one must assume that an object status changes into a non-safe state with a probability that increases with time.

Back to our example; a train that has a “fresh” MA may assume with sufficiently high probability that the route ahead is safe. With time, this information becomes stale, meaning that the probability that an object encountered underway, for instance a point, changes into a non-safe position.

This example shows that engineering data freshness is both vital and complex. Time dimensioning is highly probabilistic and should take into account the failure modes that can occur within subsystems, i.e., point detection, train coupling, rail, signal, train detection, communication, level crossing. Failure Tree Analysis can quantify the risk of any failure leading to a dangerous situation.

Most failures are detected track-side and the data models must capture the freshness threshold, which is dictated by the requirement that a failure is communicated to the train timely to avoid unacceptable danger. This analysis would take into account probabilities of:

- an object entering dangerous state
- failure being within an MA
- a train not receiving up-to-date information (includes radio-hole)
- risk exceeding Safety Integrity Level

This brief analysis suggests that detailed analysis of these factors is way beyond the scope of this project.

However, data modellers should assign to the state vector a configurable timer that starts when the vector is filled. This allows the train- and track-side to make assumptions whether the communication partner disposes of fresh data that prevent a fail-safe reaction. The value of these timers would normally be subject to statistical analysis but for all intents and purposes, this project can maintain a fair estimate given by domain experts.

7. Recommendations for Integration in CENELEC Process

Figure 47 [Winther_2008] describes the relationships between the CENELEC standards. Given that the subject of study of PERFORMINGRAIL is ERTMS Moving Block and Virtual Coupling signalling systems, given that the EURORADIO protocol is out of the scope of the project itself, and given the attention the EN50128 standard gives to the modelling concerns (e.g., semi-formal modelling, formal modelling, simulation models), we consider in this deliverable EN50128.

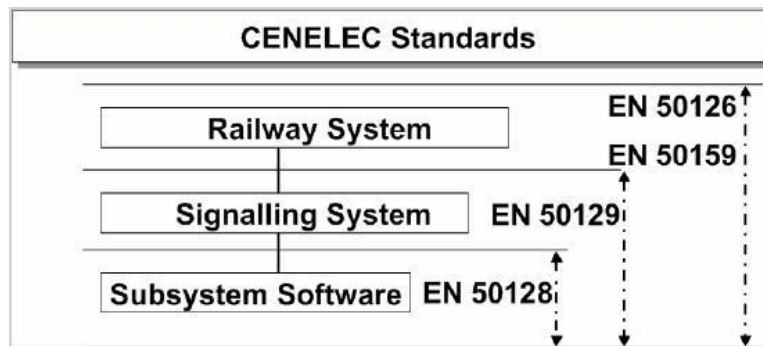


Figure 47. CENELEC standards – relationships [Winther_2008]

The following table reports a synthesis of the recommendation EN5018 prescribes on modelling with respect to the phases of system lifecycle. In such a table, recommendation levels are reported for each technique/methodology and for each SIL, according to the standards: the levels are M (mandatory), HR (high recommended), R (recommended) and NR (not recommended).

Concern	Notes
Specifications	HR for SIL3-4
Architecture Definition	HR for SIL3-4
Design & Implementation	HR for SIL1-4
Testing	R for SIL1-4

EN50128 details the level of recommendation with respect to different system aspect to model and modelling approaches:

- Finite State Machines or State Transition Diagrams are HR for all systems where SIL greater than 0; EN5018 fosters the usage of hierarchical composition (nesting and parallelism) as in UML;
- Prototyping/Animation is just classified as R;
- Data Modelling, Data Flow Diagrams, Control Flow Diagrams, Time Petri Nets, Decision/Truth Tables, Formal Methods, Performance Modelling, Structure Diagrams and Sequence Diagrams are HR for SIL3-4 and R for lower SILs.

Among formal specification approaches, EN50128 explicitly mentions: CSP, CCS, HOL, LOTOS, OBJ, Temporal Logic, VDM, Z Method, B Method and Model Checking.

The use of UML is encouraged since “[UML] facilitates the assessment of the key characteristics

of the design on the basis of representations at appropriate levels of detail. UML is frequently used in so-called model-driven development, supported by commercial products. This development style aims at improving the quality of the software and the productivity of the developers by the use of high-level modelling languages.” Hence, the standard also positively reports the use of DSML and UML Profiling as a valid technique to raise the level of abstraction of modelling tasks. Another important requirement the EN50128 prescribes is the definition and usage of a modelling guideline which considers at least one HR technique.

With respect to the recommendations of EN5018 on modelling aspects, the methodology reported in Section 4 could be considered as a valid starting point for the following reasons:

- it explicitly addresses all the phases considered by the EN5018 (from requirements engineering to model validation);
- it deals with these engineering process aspects that are considered mandatory that are applicable to the scope of this project:
 - Component - Fully Defined Interface (see MB Structural Modelling phase 2.1);
 - Design – Modular Approach (phases 2 and 3);
 - Design – Design Standard (usage of standard modelling languages);
 - Verification and Testing – Traceability (phases 3.2, 3.3, 4.3 and 4.5).

According to the recommendations of CENELEC standards, the future refinement activities of the modelling methodology, will take into account these considerations and will develop an automation approach oriented to the adoption of Model-Based Software Engineering (MBSE) in the PERFORMINGRAIL. Such refinements will explicitly address the high-level modelling languages detailing the related guidelines for the modelling of ETCS MB behaviour; the low-level languages able to provide an analysis of the system properties to evaluate; and a mapping between these two levels. Furthermore, the adoption of UML/SysML OMG standards, will surely foster the adoption of the modelling approaches and artefacts developed during the PERFORMINGRAIL project in railway industrial settings.

8. Conclusions

This deliverable presents the work that has been carried out in the PERFORMINGRAIL project, WP2, in task T2.1 and T2.2. Acknowledging the radical technological change that moving block and virtual coupling trains represent, the document presents a systematic characterization of relevant operational scenarios, potentially cross-cutting various train use cases, via a well-defined template based on established terminology with respect to performance indicators, functional components, parameters, variants, and behaviour.

On the one hand, the document paves the way towards the formalization of operational scenarios information, by facilitating semi-formal and formal modelling via the template-based description. On the other hand, the document identifies and describes the guidelines for such modelling with respect to structure, behaviour, hazards and data involved in the instantiated scenarios. The intended goal of the document is achieved by also relying on previous work carried out in former X2Rail projects, as described in the Executive Summary and subsequent sections.

A conclusion of the document is the fact that the proposed template is indeed able to describe all the selected relevant scenarios, exemplified by the instantiated template on 10 relevant MB and VC operational scenarios found in the Appendix. In addition, the details on performance indicators, parameters, variants and the behaviour itself facilitated the identification of appropriate frameworks expressive enough to capture the structure, behaviour and hazards of such systems executing the exemplified operational scenarios. The work in this document is crucial to meeting the goals of tasks T2.3 and T2.4 of WP2.

References

- [Abbors_2009] Abbors, Fredrik, Truscan, Dragos, et Lilius, Johan. Tracing requirements in a model-based testing approach. In: 2009 First International Conference on Advances in System Testing and Validation Lifecycle. IEEE, 2009. p. 123-128.
- [Abrial_1996] Jean-Raymond Abrial. The Bbook: Assigning Programs to Meanings. Cambridge University Press, New York, NY, USA, 1996.
- [Abrial_2000] Jean-Raymond Abrial. Modeling in Event-B: System and Software Engineering. Cambridge University Press, New York, NY, USA, 1st edition, 2010.
- [Abrial_2010] Jean-Raymond Abrial, Michael Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta, and Laurent Voisin. Rodin: an open toolset for modelling and reasoning in Event-B. STTT, 12(6):447_466, 2010.
- [Alexander_1977] Alexander, C., S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel. 1977. A Pattern Language: Towns – Buildings – Construction. New York, NY, USA: Oxford University Press.
- [Alur_1994] R. Alur and D. Dill: “A theory of timed automata”. Theoretical Computer Science, 126:183-235, (1994).
- [Alur_1999] R. Alur: “Timed Automata”, Procs. of 11th Int. Conf. Computer Aided Verification (1999).
- [Back_1983] Ralph-Johan Back and Reino Kurki-Suonio. Decentralization of process nets with centralized control. In Proceedings of the Second Annual ACM Symposium on Principles of Distributed Computing, PODC '83, pages 131_142, New York, NY, USA, 1983. ACM.
- [Batteux_15] Michel Batteux, Tatiana Posvirnova, Antoine Rauzy; System Structure Modeling Language (S2ML). 2015. hal-01234903.
- [Bernardi_11] Bernardi, S., Flammini, F., Marrone, S., Merseguer, J., Papa, C., Vittorini, V.; Model-driven availability evaluation of railway control systems; LNCS (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6894 LNCS, pp. 15-28, (2011).
- [Bernardi_13] Bernardi, S., Flammini, F., Marrone, S., Mazzocca, N., Merseguer, J., Nardone, R., Vittorini, V.; Enabling the usage of UML in the verification of railway systems: The DAM-rail approach; Reliability Engineering and System Safety, 120, pp. 112-126, (2013).
- [Bozzano_11] Bozzano, M., Cimatti, A., Katoen, J.-P., Nguyen, V.Y., Noll, T., Roveri, M.; Safety, dependability and performance analysis of extended AADL models; (2011) Computer Journal, 54 (5), pp. 754-775. DOI:10.1093/comjnl/bxq024
- [Bulychev_12] Bulychev P., David A., Larsen K.G., Legay A., Li G., and Poulsen D.B. Rewrite-based Statistical Model Checking of WMTL. In RV Conference. Springer, 260–275, 2012
- [David_12] David A., Du D., Larsen K.G., Legay A., Mikučionis M., Poulsen D.B., and Sedwards S. Statistical Model Checking for Stochastic Hybrid Systems. arXiv preprint arXiv:1208.3856 (2012).
- [Delange_14] Delange, J., Feiler, P.; Architecture fault modeling with the AADL error-model annex; Proc. - 40th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2014, art. no. 6928836, pp. 361-368. (2014).

[Diethers_02] K. Diethers, U. Goltz, and M. Huhn. Model checking UML statecharts with time. In *Critical Systems Development with UML, Proceedings of the UML'02 workshop*, pages 35–52. TU Munich, 2002.

[Durugbo_2013] Durugbo, Christopher. Integrated product-service analysis using SysML requirement diagrams. *Systems Engineering*, 2013, vol. 16, no 1, p. 111-123.

[D1.1] D1.1 PERFORMINGRAIL. (2021). Baseline system specification and definition for Moving Block Systems. PERFORMINGRAIL project: PERformance-based Formal modelling and Optimal tRaffic Management for movING-block RAILway signaling, deliverable D1.1.

[D2.1_ASTRAIL] D2.1 ASTRAIL. (2019). Modelling of the moving block signalling system. ASTRAIL project: “sAtellite-based Signalling and automation SysTems on RAILways along with formal method and moving block validation”, deliverable D2.1.

[D2.2_ASTRAIL] D2.2 ASTRAIL. (2019). Moving Block signalling system Hazard Analysis. ASTRAIL project: “sAtellite-based Signalling and automation SysTems on RAILways along with formal method and moving block validation”, deliverable D2.2.

[D4.1_ASTRAIL] D4.1 ASTRAIL (2019). Report on Analysis and on Ranking of Formal Methods. ASTRAIL project: “sAtellite-based Signalling and automation SysTems on RAILways along with formal method and moving block validation”, deliverable D4.1.

[D3.2_X2Rail-2] D3.2 X2Rail-2 (2019). System Architecture Specification and System Functional Hazard Analysis of the Fail-safe Train Positioning subsystem. X2Rail-2 project: "Enhancing railway signalling systems based on train satellite positioning, on-board safe train integrity, formal methods approach and standard interfaces, enhancing traffic management system functions".

[D5.1_X2Rail-2] D5.1 X2Rail-2. (2018). Formal Methods (Taxonomy and Survey), Proposed Methods and Applications, X2Rail-2 project, “Enhancing railway signalling systems based on train satellite positioning, on-board safe train integrity, formal methods approach and standard interfaces, enhancing Traffic Management System functions”.

[D1.1_MOVINGRAIL] D1.1 MOVINGRAIL. (2020). Report on Moving Block Operational and Engineering Rules. MOVINGRAIL project: “Moving block and virtual coupling, next generations of rail signalling”, deliverable D1.1.

[D4.1_MOVINGRAIL] D4.1 MOVINGRAIL. (2020). Market potential and Operational Scenarios of Virtual Coupling. MOVINGRAIL project: “Moving block and virtual coupling, next generations of rail signalling”, deliverable D4.1.

[D2.1_4SECU Rail] D2.1 4SECU Rail (2020). Specification of formal development demonstrator. 4SECU Rail project: “FORMal Methods and CSIRT for the RAILway sector”, deliverable D2.1.

[EEIG_ERTMS] EEIG ERTMS Users Group, Hybrid ERTMS/ETCS Level 3, Ref: 16E042 - 1C, 13/07/2018.

[Eulynx] <https://www.eulynx.eu/>.

[Ex_AADL] <https://github.com/osate/examples/blob/master/Train/version1/model.aadl>
[Accessed June 2021].

[Faugère_07] Faugère, M., Bourbeau, T., De Simone, R., Gérard, S.; MARTE: Also an UML profile for modeling AADL applications; (2007) Proc. of the IEEE International Conf. on Engineering of

Complex Computer Systems, ICECCS, art. no. 4276333, pp. 359-364.

[Feiler_07] Feiler, P.H., Lewis, B.A., Vestal, S.; The SAE architecture analysis & design language (AADL) a standard for engineering performance critical systems; (2007) Proc. of the 2006 IEEE Conference on Computer Aided Control Systems Design, CACSD, art. no. 4064767, pp.1206-1211.

[Flammini_2014] F. Flammini, S. Marrone, M. Iacono, N. Mazzocca and V. Vittorini: "A multiformalism modular approach to ERTMS/ETCS failure modeling", Int. Journal of Reliability, Quality and Safety Engineering, vol. 21, No. 01, (2014).

[Flammini_21] F. Flammini, S. Marrone, R. Nardone, and V. Vittorini: "Compositional Modeling of Railway Virtual Coupling with Stochastic Activity Networks", accepted for publication, Formal Aspects of Computing, Springer.

[França_07] França, R.B., Bodeveix, J.-P., Filali, M., Rolland, J.-F., Chemouil, D., Thomas, D.; The AADL behaviour annex - Experiments and roadmap; Proc. of the IEEE Int. Conf. on Engineering of Complex Computer Systems, ICECCS, art. no. 4276336, pp. 377-382, (2007).

[Friedenthal_2006] Friedenthal, Sanford, MOORE, Alan, et STEINER, Rick. OMG systems modeling language (OMG SysML) tutorial. In: INCOSE Intl. Symp. 2006. p. 65-67.

[Gehlot_2010] Gehlot, Vijay, and Carmen Nigro. "An introduction to systems modeling and simulation with colored petri nets." *Proceedings of the 2010 winter simulation conference*. IEEE, 2010.

[Get_08] Get, J.H., Get, B.Z., Get, L.P., Kordon, F.; From the prototype to the final embedded system using the Ocarina AADL tool suite; Trans. on Embedded Computing Systems, 7 (4), art. no. 42, (2008).

[GoF_1995] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design patterns: elements of reusable object-oriented software. Addison-Wesley Longman Publishing Co., Inc., USA, 1995.

[Hause_2006] Hause, Matthew, et al. The SysML modelling language. In : Fifteenth European Systems Engineering Conference. 2006. p. 1-1

[Heitmeyer_94] Heitmeyer C. and Lynch, N. The generalized railroad crossing: a case study in formal verification of real-time systems, in Proceedings of Real-Time Systems Symposium, San Juan, Puerto Rico, USA, 1994, pp. 120-131

[Henzicher_1996] Henzicher, T. A. The theory of hybrid automata, LICS'96: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (Washington, DC, USA). IEEE Computer Society, 1996, p. 278.

[IEEE1471] 1471-2000 - IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. 2000.doi:10.1109/IEEESTD.2000.91944. ISBN 0-7381-2518-0.

[ISO_29148] International Standard ISO/IEC/IEEE 29148, Systems and software engineering — Life cycle processes – Requirements engineering, 1st edition, 2011-12-01

[ISO/IEC/IEEE 42010] "ISO/IEC/IEEE 42010:2011 - Systems and software engineering - Architecture description". International Organization for Standardization. 2011-11-24.

[Jensen_1981] Jensen, Kurt. "Coloured Petri nets and the invariant-method." *Theoretical computer science* 14.3 (1981): 317-336.

- [Jensen_2007] Jensen, Kurt, Lars Michael Kristensen, and Lisa Wells. "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems." *International Journal on Software Tools for Technology Transfer* 9.3 (2007): 213-254.
- [Kruchten_1995] Kruchten, Philippe; Architectural Blueprints — The “4+1” View Model of Software Architecture. *IEEE Software* 12 (6), pp. 42-50 (1995).
- [Leveson_2004] Leveson N., A new accident model for engineering safer systems, *Safety Science journal*, vol. 42, no. 4, pp. 237-270, 2004.
- [Murata_1989] Murata, T.: "Petri nets: Properties, analysis and applications," in *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541-580 (1989).
- [Marsan_1995] Marsan, M.A, Balbo, G., Conte, G., Donatelli, S., and Franceschinis, G.: “*Modelling with Generalized Stochastic Petri Nets*”, Wiley Series in Parallel Computing, John Wiley and Sons (1995).
- [MARTE] OMG; UML Profile for MARTE; Version: 1.2; April 2019.
- [MARTE-DAM] Bernardi, S., Merseguer, J., Petriu, D.C.; A dependability profile within MARTE; *Software and Systems Modeling*, 10 (3), pp. 313-336, (2011).
- [Ölveczky_10] Ölveczky, P.C., Boronat, A., Meseguer, J.; Formal semantics and analysis of behavioral AADL models in real-time Maude; *LNCS (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6117 LNCS, pp. 47-62, (2010).
- [OMG_2006] Object Management Group (OMG), 2006, Systems Modeling Language version 1.0, Available from <https://www.omg.org/spec/SysML/1.0/PDF>. [Accessed June 2021].
- [Ravikumar_2016] Ravikumar S., Subramaniam C., A Survey on Different Software Safety Hazard Analysis and Techniques in Safety Critical Systems, *Middle-East Journal of Scientific Research*, Issue 24 (Special Issue on Innovations in Information, Embedded and Communication Systems): 90-97, 2016.
- [Richards_20] Mark Richards, Neal Ford; *Fundamentals of Software Architecture*; O'Reilly Media, Inc.; January 2020.
- [SAF] Alexander Haarer, Klaus Rödel, Christian Lalitsch Schneider; *System Architecture Framework (SAF)*, GfSE Arbeitsgruppe SAF – Incose.
- [Sanders_2001] Sanders, W., Meyer, J.: *Stochastic activity networks: “Formal definitions and concepts”*. *Lecture Notes in Computer Science* 2090, 315–343 (2001).
- [SCADE] Güdemann, Matthias & Ortmeier, Frank & Reif, Wolfgang. (2007). Using Deductive Cause Consequence Analysis (DCCA) with SCADE. 10.1007/978-3-540-75101-4_44.
- [Simulink] Nitsch, A., Beichler, B., Golatowski, F., & Haubelt, C. (2015). *Model-based Systems Engineering with Matlab/Simulink in the Railway Sector*. MBMV.
- [Soares_2011] Dos Santos Soares, Michel, VRANCKEN, Jos, et VERBRAECK, Alexander. User requirements modeling and analysis of software-intensive systems. *Journal of Systems and Software*, 2011, vol. 84, no 2, p. 328-339.
- [Simon_2011] Simon Hudon and Thai Son Hoang. Development of Control Systems Guided by Models of their Environment. In *Electronic Notes in Theoretical Computer Science*, Volume 280, pages 57-68, 2011.

[SYSML19] OMG. Systems Modeling Language - Version 1.6; OMG Document Number: formal/19-11-01; November 2019; <https://www.omg.org/spec/SysML/1.6/>

[UML] OMG. Unified Modeling Language. <https://www.omg.org/spec/UML/2.1.2>, 2007.

[UML02] OMG. Unified Modeling Language Superstructure - Version 2.2; OMG Document Number: formal/2009-02-02; <http://www.omg.org/spec/UML/2.2/Superstructure>

[UNISIG_026] UNISIG, ERTMS/ETCS - System Requirements Specification, Chapter 5 Procedures. SUBSET-026-5, 3.6.0, 13/05/2016

[Vanit_2018] Vanit-Anunchai, Somsak. "Modelling and simulating a Thai railway signalling system using Coloured Petri Nets." *International Journal on Software Tools for Technology Transfer* 20.3 (2018): 243-262.

[Weilkiens_07] Weilkiens, T.; Systems Engineering with SysML/UML; (2007).

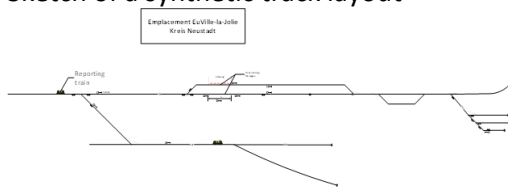
[Winther_2008] Troels Winther, Quick Guide to Safety Management based on EN 50126 / IEC 62278, <https://en50126.blogspot.com/2008/07/>. [Accessed June 2021].

Appendix A – OS#1 - Trackside initialisation

Operational Scenario #1	
Title:	Trackside initialisation
Abstract:	This scenario describes the process of initialising the trackside control systems with up-to-date values.
Description:	<p>The concept of state vector and its initialisation is central to this scenario. Trackside in this context is the area under control of an RBC or more general, of a Central Safety System that combines RBC and interlocking.</p> <p>Trackside area is an area that can be located, both geographically and in terms of track topology.</p> <p>State space represents the status of the trackside system. This is represented by a state vector, i.e. a vector of (object, state)-tuples. Objects include fixed trackside elements such as points as well as transient objects such as trains and temporary speed restrictions.</p> <p>State vector initialisation allocates state to the values. State is detected through sensors, actuators and messages.</p>

Applicable Use Case(s)	
1	Normal train movement
2	Staff responsible (SR) movement

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Startup time	Quantitative	Availability	1-15 minutes	The time the system needs to reach operational status
Completeness	Quantitative	Reliability, Safety	10e-2	The probability that an object-status remains unknown (null)
Safety	Quantitative	Safety	10e-9	The probability that a vital object state value is detected wrongly

Signalling Type	System Type	Track Information
General	Full MB (FMB)	<p>Sketch of a synthetic track layout</p> 

Functional components
L3 Trackside
Traffic Management System

Trackside Train Detection
Adjacent Signalling System
Train/TIMS
Dispatcher
Driver

Trackside Function(s)	ETCS On-Board Function(s)
Track Status Management	Integrity Information Management
Points Management	Train position reporting
Reserved Status Management	

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)	Acquisition time-out	Order 1 minute	Time after which objects are considered non-reporting	
Train	Equipped train		Train equipped with ETCS L3 and TIMS	ETCS subset-026
	Yellow fleet		Maintenance trains	
Speed	V_t		Trains are assumed to be stationary during trackside initialisation	
Track	Track configuration dataset	XML data	See [EULYNX, 2020]	Dataprep.eulynx.eu
Position	Pos_t		Position in terms of LRBG and mapped to the track topology	ETCS subset-026

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	Track assets and trains in the control area are void	State vector fully known (Track-Circuit, Virtual Sub-Sections, etc.)	Trackside system reboot	Static track configuration is known. Vital transient objects such as trains, TSR, MA's are persisted in non-volatile memory.
Desc.				
	#1	Central safety system is powered off.		
	#2	Trains stop receiving life-signs and apply fail-safe reaction.		
	#3	Trackside signals are at danger (dark equates to showing red, i.e. at danger).		
	#4	Train status updates are lost		
A	Track-train comm ceased	Trains stationary	Comm time-out on application level	Trains will not exceed MA. No yellow fleet movement.
Desc.	Trains brake to standstill as EoA or trainside lifesign supervision times out			
	#5.A	Set of trains in control area slow down yet remain within MA area.		

	#6.A	Train enters the control area using the MA received before trackside went down.		
B	Procedure allows staff to reboot the systems	No system failures	Systems re-energised	All trains stopped within area of stored MA
Desc.	Trackside system powered			
	#5.B	Trackside object status acquisition		
C	Field elements can report status	All field element stati filled	Comm with field elements resumed	
Desc.	Trackside object status acquisition, e.g. detection of points			
	#5.C	Trackside elements report status		
	#6.C	Trains allocated to a "dead-reckoning position window" based on stored MA		
D	Train-Track contact re-established	Vital train positions known	Trains contact RBC	Vital train front- and rear-end known. No yellow fleet movement.
Desc.	Train status acquisition			
	#5.D	Trains report stored position		
	#6.D	Trains report integrity		
	#7.D	Trackside updates positions and resolves train separation		
E	Field elements and train positions detected		RBC system release timers expired	No stale entries in state vector
Desc.	Trackside calls routes and the RBC is released, meaning that it can issue MA's			
	#5.E	Signaller or ARS calls route		
	#6.E	IL locks route and RBC issues MA		
	#7.E	Trains receive MA		

Variant	Description	Alternatives	Main case	Impact/Affected Steps
V1	TTD at points so that point sections can be detected vacant after reboot	Track circuits or axle counters.		7B, ability to detect point sections reduces the effect of unknown vacancy status spilling over to the rest of the network
V2	Non-reporting train after reboot			11.D and later. RBC can't fully inform track vacancy status reducing scope for MA calculation.
V3	L3 signalling overlaid on legacy NTC system	Legacy NTC system is spot (i.e. balise based) - or continuous (e.g. loop or radio-based)		Legacy systems rely on the presence of TTD so this precludes FMB
V4	Propagation timer expires before initialisation completes. This invalidates stored train positions.			Creates the need to sweep large swathes of track that have unknown occupancy status.

Hazards		
ID	Description	Reference/new possible hazard
H-Clearing-001	Track status area erroneously cleared; manual override by signaller	[Beugin, 2021] §3.8.1
H-Clearing-002	Track status area erroneously cleared; invalid/outdated system information	[Beugin, 2021] §3.8.1
H-data-entry-001	Persistent storage of transient objects (TSR, slippery track, train position) is lost. Manual re-entry of data is wrong.	[EULYNX, 2020] Figure 3.2, Track status information mismatch

ID	Applicable Operational Rules	Reference
H-Clearing-002a	Yellow fleet movement during initialisation phase	Track status information mismatch
H-Clearing-002b	Driver shuts down ETCS equipment and moves train beyond bounds of the MA during the initialisation phase.	Track status information mismatch

ID	Applicable Requirements	Reference

GLOSSARY

Name	Type	Description	Acronym (if any)
Central safety system	functional	Interlocking and RBC	CSS
ARS	Functional	Automatic Route Setting	
Yellow fleet	Functional	Maintenance rail vehicles	
TC	Functional	Track Circuit	
MA	Functional	Movement Authority	
VSS	Functional	Virtual SubSection	
TTD	Functional	Trackside Train Detection	

REFERENCES

[D4.2 X2R3] D4.2–Moving Block System Specifications, X2Rail-3 project, Advanced Signalling, Automation and Communication System, deliverable D4.2, part 1 to 6, 18/12/2020.

[Beugin, 2021] Beugin, J., 2021. Synthesis report on the state-of-the-art of Moving Block system specifications and available technologies, s.l.: s.n.

[EULYNX, 2020] EULYNX Data Preparation. [Online]
Available at: <https://eulynx.eu/index.php/dataprep>

Appendix B – OS#2 - Start of Mission

Operational Scenario #2	
Title:	Start Of Mission
Abstract:	This scenario concerns the Start of Mission (SoM) of a non-localised train followed by Staff Responsible to re-locate the train. When the driver begins the Start of Mission procedure, the train PR status is “Unknown” and L3 Trackside authorizes the train to run in Staff Responsible mode until the train reaches a first location reference and reports its position to the L3 Trackside.
Description:	<p>General context of SoM:</p> <p>A stopped train in Stand-by (SB) mode and under the supervision of the L3 system has to start a mission with a SoM procedure to reach Full Supervision (FS), On-Sight (OS), Limited Supervision (LS), or Staff Responsible (SR) modes. When the SoM procedure is launched, the train cab desk is assumed to be already open by the driver and, no communication session is currently established or being established between the on-board and trackside parts. Main steps of the procedure are:</p> <ol style="list-style-type: none"> 1. Opening of a communication session 2. On-board validation of stored data 3. Position status determination 4. Mode selection <p>Assumption for this scenario: steps 1 and 2 are successful, step 4 is SR.</p> <p>In terms of performance indicator evaluation, this operational scenario may allow for the analysis of the success/failure to obtain a first correct and valid position after the SoM procedure. This can be especially analysed when the SoM procedure is started somewhere on a line equipped only with virtual balises. VB hazards are related to GNSS-based VB reader issues, in particular issues related to GNSS feared events.</p>

Applicable Use Case(s)	
1	Start of Train (=Start of Mission)
2	Movement in SR mode
3	Loss of Train Integrity

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Probability of the first position to be erroneous while L3 trackside receives a valid PR (i.e. format correct but real position)	Quantitative	Safety		Particular attention is put on the first train position because this position is not protected by the “Linking” mechanism, as it is the case for further positions. A wrong first train

wrongly bounded)				position can lead to a wrong first MA and, eventually to a potential train collision.
------------------	--	--	--	---

Signalling Type	System Type	Track Information
General	Full MB (FMB)	

Functional components
L3 Trackside
ETCS On-board
Traffic Management System
Train/TIMS
Driver

Trackside Function(s)	ETCS On-Board Function(s)
Communication Management	Train Position Reporting
Trains Management	Integrity Information Management
Track Status Management	Speed and Distance Supervision

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)	SESSION_TIMER	Max=5min		[SS026 part 3-Appendix A.3.1]
	VALID_DATA_TIMER	To be defined	Timer with a defined timeout for receiving Validated Train Data.	[X2R3 D4.2]
Train (in [Msg136] / [Msg157]: Position Report / SoM PR) LRBG info used in SoM are those of the previous mission	NID_LRBG	[0 ... 2 ²⁴]	Identity of last relevant balise group	[SS026 part7]
	D_LRBG	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	Distance between the last relevant balise group and the estimated front end of the train	[SS026 part7]
	Q_DIRLRBG	0: Reverse 1: Nominal 2: Unknown	Qualifier for the orientation of the train in relation to the direction of the LRBG	[SS026 part7]
	Q_DLRBG	0: Reverse 1: Nominal 2: Unknown	Qualifier telling on which side of the LRBG the estimated front end is	[SS026 part7]

	L_DOUBTOVER = MaxSFE	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	over-reading amount (odometry error + error in detection of balises in rear of the estimated train position) + Q_LOCAC (position error in meter) of the LRBG location	[SS026 part7]
	L_DOUBTUNDER = MinSFE	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	under-reading amount (odometry error + error in detection of balises in front of the estimated train position) + Q_LOCAC (position error in meter) of the LRBG location	[SS026 part7]
	Q_LENGTH (Linked to CRE)	0: No TI information available 1: TI confirmed by TIMS 2: TI confirmed by driver 3: TI lost	Qualifier for train integrity status	[SS026 part7]
	L_TRAININT (Linked to CRE)	[0 ... 327 667] unit: m resolution: 1m	Safe Train length	[SS026 part7]
	V_TRAIN	[0 ... 600] unit: km/h resolution: 5 km/h	Train speed	[SS026 part7]
	Q_DIRTRAIN	0: Reverse 1: Nominal 2: Unknown	Qualifier for the direction of train movement in relation to the LRBG orientation	[SS026 part7]
	M_MODE	0: FS 1: OS 2: SR 3: SH 6: SB 12: LS	On-board operating mode	[SS026 part7]
	NID_LRBG	[0 ... 2 ²⁴]	Identity of last relevant balise group	[SS026 part7]

Train (in [Msg157]: SoM PR)	Q_STATUS	0: Invalid 1: Valid 2: Unknown	Status of SoM position report	[SS026 part7] [SS026 part8]
Train (in [Msg129]: Validated Train Data)	L_TRAIN	[0 ... 4095] Unit: m resolution: 1m	This is the absolute real length of the train.	[SS026 part7] [SS026 part8]
Train	T_TRAIN	[0 ... 42949672.94] Unit: s resolution: 10ms	Timestamp linked to any information, i.e. time, according to trainborne clock, at which message is sent	[SS026 part7] [SS026 part8]
Speed	V0	0 for SoM	Initial train speed	
Track (in [Msg2])	D_SR	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	Distance that can be covered in SR mode	[SS026 part7] [SS026 part8]
Track Stored info	Train location interval	[Min _{TLI} ... Max _{TLI}]	The TLI at a given time t is between MaxSFE at t and the last known rear end, i.e. CRE at t = L_DOUBTOVER – L_TRAININT For SoM, if Q_STATUS = “valid”, the last known rear end = MinSRE at t = L_DOUBTOVER – L_TRAIN	

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	<p>Initial state of L3 Trackside:</p> <ul style="list-style-type: none"> - The L3 Trackside has stored an "Unknown" track section linked to the previous location of the train (when the train driver performed an EoM). - The communication session is considered as closed and the train disconnected. <p>Initial state of ETCS On-Board:</p> <ul style="list-style-type: none"> - EVC is in SB mode (the desk has been opened, or the driver has selected Exit from Shunting) - EVC has requested that the driver enters/revalidates his Driver ID. - The level stored is L3, Train Running Number and RBC data are valid. - EVC is able to consider Train Position status. 			
Desc.	The ETCS On-board and the L3 trackside establish a new communication session , then the ETCS On-board sends train data to the L3 trackside. The steps are described for the following operational contexts: Terminal Station/ Intermediate Station/ In Line.			
Sub. Desc.	New communication session: the On-board starts from the safe connection set-up until the sending of the SoM Position Report (excluded)			
	#1	The On-board EVC sends the message Initiation of Communication Session to the RBC [Msg155] that checks that the number of connected trains is lower than the maximum number of train connections managed by RBC (configuration parameter).		
	#2	RBC sends the RBC/RIU System Version [Msg32] to the EVC with the request of acknowledgement (M_ACK=1) and M_VERSION = 32 (Baseline 3-Release 2)		

#3	EVC sends the ACK message [Msg146] to the message [Msg32], and starts the verification of the version compatibility between trackside and EVC.			
#4	RBC receives the ACK message [Msg146] and starts the SESSION_TIMER waiting to receive the message Session Established [Msg159].			
#5	EVC verifies the compatibility and sends the Session Established message [Msg159]. Assumption: The system versions are compatible, the EVC considers the communication session established for the On-Board and informs the driver.			
#6	RBC receives the Session Established message [Msg159], stops the SESSION_TIMER and considers the communication session also established for the Trackside.			
Desc.	Sending of train data: the EVC sends a SoM Position Report with the Train Position status and the previous status of stored information ¹²			
#7	The EVC sends the SoM PR message [Msg157], including the information on the stored position (Q_STATUS) and the Train Running Number (NID_OPERATIONAL) [Pkt5].			
#8	RBC receives the SoM Position Report message [Msg157] and checks: - if PR variables have some "Unknown" values - if the LRBG of the PR is known and is included in the RBC configuration			
A		RBC sends a MA with an OS or FS profile		
Desc.	L3 Trackside receives the SoM PR with a "Valid" status (Q_STATUS = 1) and receives the other information stored on-board with a "Valid" status			
#9.A	RBC considers the SoM PR as "Valid" and the L3 Trackside sends to the train the information to select the track in a non-ambiguous way where the train is localised. VALID_DATA_TIMER is started. Context: The reference BG of the PR received is located in advance (for the Terminal Station context) / in rear (for the Intermediate Station or In Line context) of the Min Safe Front End (MinSFE) of the train. There is no switch point that may lead to an alternative route, located between the LRBG and the MaxSFE (with respect to the train orientation).			
#10.A	EVC sends the Validated Train Data message [Msg129] to RBC, including the Position Report packet [Pkt0] and the Validated train data packet [Pkt11].			
#11.A	RBC receives the message [Msg129], checks the data, considers such data as acceptable, VALID_DATA_TIMER is stopped, and a handshake takes place (RBC sends the ACK of Train Data message [Msg8] to EVC, EVC sends the ACK message [Msg146] to RBC, RBC receives the ACK message [Msg146])			
#12.A	Driver interaction: The START button is enabled on DMI. The Driver waits to receive from the Dispatcher the authorization to press it (it depends on conditions of freedom of the in advance of the train position), if so, the START button is pressed. On-board event: The EVC sends the MA Request message [Msg132] and waits for the authorization to move.			

¹² From [ERSAT-GGC D2.1][§5.4.1] [§5.4.3][§5.5]

	#13.A	RBC sends the MA message [Msg3] to EVC (or the MA with the Shifted Location Reference message [Msg33] for Terminal Station context): - An On-Sight (OS) or a Limited Supervision (LS) profile [Pkt80] up to the first (virtual?) signal in advance of the train - Full Supervision Movement Authority starting from the first (virtual?) signal in advance of the train up to the last (virtual?) signal where the FS conditions are all fulfilled		
B		RBC sends a SR authorization and waits for the train to reach a first location reference and to report it.		
Desc.	L3 Trackside receives the SoM PR with an “Unknown” status (Q_STATUS = 2) and receives the other information stored on-board with a “Valid” status. In this case, the train is considered as not localised but the L3 Trackside accepts it. Assumption: L3 Trackside has stored the association between NID_ENGINE and Track section of the train, and it knows the direction of the train. ¹³			
	#9.B	RBC regards the SoM PR as “Unknown” and, then, RBC considers the train not localised. RBC starts the handshake to inform the EVC that the train has been accepted (RBC sends the ACK of Train Accepted message [Msg41] to EVC, EVC sends the ACK message [Msg146] to RBC, RBC receives the ACK message [Msg146]). VALID_DATA_TIMER is started.		
	#10.B	EVC sends the Validated Train Data message [Msg129] to RBC, including the Position Report packet [Pkt0] and the Validated train data packet [Pkt11].		
	#11.B	RBC receives the message [Msg129], checks the data, considers such data as acceptable, VALID_DATA_TIMER is stopped, and a handshake takes place (RBC sends the ACK of Train Data message [Msg8] to EVC, EVC sends the ACK message [Msg146] to RBC, RBC receives the ACK message [Msg146])		
	#12.B	Driver interaction: The START button is enabled on DMI. The Driver waits to receive from the Dispatcher the authorization to press it (it depends on conditions of freedom of the in advance of the train position), if so, the START button is pressed.		
	#13.B	RBC checks that a not connected train is present on the track section and the association between NID_ENGINE stored and NID_ENGINE received from the EVC is the same. RBC sends to the train the information to select the track/the platform in a non-ambiguous way, according to the information stored at L3 Trackside. The train position is considered “Approximate” by RBC.		
	#14.B	The EVC sends the MA Request message [Msg132] and waits for the authorisation to move.		
	#15.B	RBC sends an SR Authorisation message [Msg2] to EVC with D_SR=infinite and including the List of Physical/Virtual Balises in SR Authority packet [Pkt63] and including Physical/Virtual Balise of the current Radio Block Section plus the Balise of the RBS in advance.		
	#16.B	The driver acknowledges and the train starts to move in SR mode.		

¹³ From [ERSAT-GGC D2.1][§5.6.1]

#17.B	The Train detects the first Physical/Virtual BG and EVC sends a Position Report message [Msg136].
-------	---

Variant	Description	Alternatives	Main case	Impact/Affected Steps
V1	<p>This is a variant of Branch B when RBC is not able to associate the track with train NID_ENGINE because this information is not stored at L3 Trackside (cf. assumption of B).</p> <p>There is a small change in the behaviour description but the post-condition remains the same.</p>	<ul style="list-style-type: none"> - V1.A: this information can be received by RBC from TMS (or a dispatcher command) - V1.B: If there is no TMS/RBC connexion, the train position cannot be considered as "Approximate", in that case, the dispatcher authorises the Driver to start the Override procedure 		In V1.B => #14.B and #15B are skipped
V2	This variant considers train integrity problems.	<ul style="list-style-type: none"> - V2.A: Previous train length stored at L3 trackside and current train length are different - V2.B: Train has not confirmed its integrity (Q_LENGTH = "No TI information available") 		The train is or is not authorised to move. If it is, the track status is updated accordingly by increasing the initial unknown area linked to the train.

Hazards		
ID	Description	Reference/new possible hazard
H-StartTrain-001	<ul style="list-style-type: none"> - Degraded case of SoM with the In Line context - SoM PR with an "Unknown" status - The next location reference encountered by the train in SR mode can only be a Virtual Balise - The first reported position based on VB is wrong due to problem in the GNSS-based VB reader. 	[ERSAT-GGC D2.1] [§5.6.7] and [ERSAT-GGC D3.2] [§11.3]

ID	Applicable Operational Rules	Reference
OPE-Generic-3	When asked by the Dispatcher to report the location of the train, the Driver shall do so in accordance with non-harmonised rules.	[X2R3 D4.2]
OPE-StartTrain-1	At Start of Mission or following a change in train length (e.g. splitting and joining), the Driver shall check that the Train Integrity Monitoring System (TIMS), where fitted, is operational.	[X2R3 D4.2]
OPE-StartTrain-	Non-harmonised Operational Rules shall define the circumstances	[X2R3 D4.2]

2	under which the Driver is allowed to move a train unable to report a confirmed integrity.	
OPE-StartTrain-3	Non-harmonised Operational Rules shall define where, after receiving notification of a train reporting a position which cannot be determined by the L3 Trackside, the Dispatcher and Driver shall make contact to determine an approximate position for the train.	[X2R3 D4.2]
OPE-StartTrain-4	Non-harmonised Operational Rules shall define where, after determining an appropriate position of the train with the Driver, the Dispatcher shall enter the location into the L3 Trackside.	[X2R3 D4.2]
OPE-StartTrain-5	The Dispatcher shall only authorise Override to be used for a train without a known location in the L3 trackside, after having assigned a location for this train.	[X2R3 D4.2]

ID	Applicable Requirements	Reference
REQ-StartTrain-1	The L3 Trackside shall always accept a train during Start of Mission.	[X2R3 D4.2]
REQ-StartTrain-2	The L3 Trackside shall alert the TMS of a train that, in the Start of Mission position report, it is reporting an invalid or unknown position, or a valid position from an NID_BG not known to the L3 Trackside.	[X2R3 D4.2]
REQ-StartTrain-3	The L3 Trackside shall accept from the TMS a position assigned by the Dispatcher for a train which is reporting an invalid or unknown position, if this position lies within an existing Unknown Track Status Area.	[X2R3 D4.2]
REQ-StartTrain-4	The L3 Trackside shall alert the Dispatcher via the TMS about an approximate position assigned by the Dispatcher for a train which is reporting an invalid or unknown position, if this position lies outside an existing Unknown Track Status Area.	[X2R3 D4.2]
REQ-StartTrain-5	When accepting a position assigned for a train from the TMS, the L3 Trackside shall associate the train with the Unknown Track Status Area in which the train is positioned.	[X2R3 D4.2]
REQ-StartTrain-6	The L3 Trackside shall compare the new train data and train location information reported by a train performing Start of Mission against the information stored for the same location for a previous train.	[X2R3 D4.2]
REQ-StartTrain-7	If the L3 Trackside determines that the train location information from a train after performing Start of Mission does not match the stored information associated with the Unknown Track Status Area in which the train is located, then the L3 Trackside shall remove the Unknown corresponding to the area from the Min Safe Front End to the Max Safe Rear End of that train.	[X2R3 D4.2]
REQ-StartTrain-8	If a train, which has not yet sent Validated Train data, reports a position with both the Min Safe Front End and the Max Safe Front End in an area of track considered Clear, then the L3 Trackside shall create an Unknown Track Status Area for the front end of this train from the reported Min Safe Front End to Max Safe Front End.	[X2R3 D4.2]
REQ-StartTrain-9	If a train, which has not yet sent Validated Train data, reports a	[X2R3 D4.2]

	position with a Confidence Interval which is partly in an Unknown Track Status Area and the Estimated Front End outside this Unknown Track Status Area, then the L3 Trackside shall extend this Unknown Track Status Area to the boundary of the reported Confidence Interval for the Estimated Front End of this train.	
REQ-StartTrain-10	When receiving Validated Train Data from a train which is not associated with an Unknown Track Status Area except for the Estimated Front End of this train, then the L3 Trackside shall extend this Unknown Track Status Area with the reported Train Length to the Min Safe Rear End.	[X2R3 D4.2]
REQ-StartTrain-11	If the L3 Trackside receives Validated Train Data from a train with a position within an Unknown Track Status Area for which the stored train length is less than what was reported by this train, then the L3 Trackside shall alert the TMS to the situation.	[X2R3 D4.2]
REQ-StartTrain-12	The L3 Trackside shall maintain the communication session with a train reporting an LRBG at Start of Mission with an ID set to 'unknown' or an ID which is not known to the L3 Trackside, unless requested by the TMS to terminate the session with this train.	[X2R3 D4.2]
REQ-StartTrain-13	On request from the TMS, the L3 Trackside shall order the train to terminate the communication session.	[X2R3 D4.2]
REQ-StartTrain-14	The L3 Trackside shall, if configured, alert the TMS of a train which terminated its communication session without sending Validated Train Data to the L3 Trackside.	[X2R3 D4.2]
REQ-StartTrain-15	The L3 Trackside shall, if configured, alert the TMS about communicating trains for which the L3 Trackside has not received Validated Train Data after a defined timeout.	[X2R3 D4.2]

REFERENCES

[ERSAT-GGC D2.1] D2.1—Enhanced Functional ERTMS Architecture Capable of using GNSS and Public Radio TLC Technologies, ERSAT-GGC project, ERTMS on Satellite-Galileo Game Changer 19/06/2018.

[ERSAT-GGC D3.2] D3.2—GNSS Quantitative Analysis for ERSAT GGC Project, ERTMS on Satellite-Galileo Game Changer 29/10/2018.

[SS023] ERTMS/ETCS Subset 023—Glossary of Terms and Abbreviations, Issue 3.1.0, 12/05/2014.

[SS026 part3] ERTMS/ETCS Subset 026—System Requirements Specification, Chapter 3: Principles, Issue 3.6.0, 13/05/2016.

[SS026 part7] ERTMS/ETCS Subset 026—System Requirements Specification, Chapter 7: ERTMS/ETCS language, Issue 3.6.0, 13/05/2016.

[SS026 part8] ERTMS/ETCS Subset 026—System Requirements Specification, Chapter 8: Messages, Issue 3.6.0, 13/05/2016.

[X2R3 D4.2] D4.2–Moving Block System Specifications, X2Rail-3 project, Advanced Signalling, Automation and Communication System, deliverable D4.2, part 1 to 6, 18/12/2020.

Appendix C – OS#3 - Points Control

Operational Scenario #3	
Title:	Points Control
Abstract:	This scenario concerns the moving, locking and releasing of points related to two subsequent trains requesting to pass over different points.
Description:	<p>In this scenario, the situation is considered in which two trains running under normal moving block conditions cross a point consecutively, with the second train requiring the point to move to a different position.</p> <p>This point cannot be moved as long as the first train occupies the associated track area. Also, the point cannot be moved when the point is already reserved to the second train. These point movement timing restrictions apply due to the hazard of moving a point while a train is passing, or about to pass, over it, possibly leading to derailment of the train.</p>

Applicable Use Case(s)	
1	Points control
2	Normal train movement
3	Sweeping

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Headway time	Quantitative	Performance		Time between train heads over points

Signalling Type	System Type	Track Information
General	Full MB (FMB)	

Functional components	
L3 Trackside	
ETCS OnBoard	
Traffic Management System	
Dispatcher	
Driver	

Trackside Function(s)	ETCS On-Board Function(s)
Points Management	Train Position Reporting
Track Status Management	Integrity Information Management
Reserved Status Management	Speed and Distance Supervision
Trains Management	
Route Management	
MA Management	

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)	Track section timer	To be defined		
Train	L_TRAIN	To be defined	Length of the train	
	M_MODE	0: FS 1: OS 2: SR 3: SH 6: SB 12: LS	On-board operating mode	
Speed	V_TRAIN	To be defined	Train speed	
Track	Track configuration: location of points	To be defined		
	Point control processing time (Interlocking)	To be defined		

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	<ul style="list-style-type: none"> - Track / Reserved Status of track section containing point is 'Occupied' / 'Not Reserved' (by train localisation). - Point is locked. 	<ul style="list-style-type: none"> - Track / Reserved Status of track section containing point is 'Clear' / 'Not Reserved'. - Point is set in other position. 	Movement Authority request requiring point movement.	
A	<ul style="list-style-type: none"> - Track / Reserved Status of track section containing point is 'Occupied' / 'Not Reserved' (by train localisation). - Point is locked. 	<ul style="list-style-type: none"> - Track / Reserved Status of track section containing point is 'Clear' / 'Not Reserved'. 		
Desc.	Track section containing locked point is released after Train 1 has run over the point.			
	#1.A	Trains Management receives Train Position Report of Train 1.		
	#2.A	Trains Management sends information of track section release by Train 1 to Track Status Management.		
	#3.A	Track Status Management sets status of track section to 'Clear'.		
	#4.A	Track Status Management reports release of track section to Route Management.		
	#5.A	Route Management releases route section containing the point.		

B	- Track / Reserved Status of track section containing point is 'Clear' / 'Not Reserved'.	- Track / Reserved Status of track section containing point is 'Clear' / 'Reserved'. - Point is locked in other position.		
Desc.	Point is moved and locked, and track section containing the point reserved after Movement Authority request for Train 2.			
	#1.B	Trains Management receives Train Position Report of Train 2.		
	#2.B	Trains Management reports Train Position to Traffic Management System.		
	#3.B	Route Management receives Movement Authority request for Train 2 from Traffic Management System.		
	#4.B	Route Management requests points position to Points Management.		
	#5.B	Points Management requests status of track section containing points to be moved from Track / Reserved Status Management.		
	#6.B	Track / Reserved Status Management reports 'Clear' / 'Not Reserved' status of track section containing points to be moved to Points Management.		
	#7.B	Points Management moves point.		
	#8.B	Points Management locks point position.		
	#9.B	Points Management reports locked point to Route Management.		
	#10.B	Route Management locks route.		
	#11.B	Route Management requests track section to be reserved by Reserved Status Management.		
	#12.B	Reserved Status Management sets the status of the track section to 'Reserved'.		
	#13.B	Reserved Status Management reports Reserved Status of track section to Route Management.		
	#14.B	Route Management reports locked route to Movement Authority Management.		
	#15.B	Movement Authority Management creates Movement Authority.		
	#16.B	Movement Authority Management sends Movement Authority to ETCS On-Board.		
C	- Track / Reserved Status of track section containing point is 'Clear' / 'Reserved'.	- Track / Reserved Status of track section containing point is 'Occupied' / 'Not Reserved'.		
Desc.	Track section containing locked point is occupied by Train 2.			
	#1.C	Trains Management receives Train Position Report of Train 2.		
	#2.C	Trains Management sends information of track section occupancy to Track / Reserved Status Management.		
	#3.C	Track / Reserved Status Management sets track section status to 'Occupied' / 'Not Reserved'.		
D	- Track / Reserved Status of track section containing point is 'Occupied' / 'Not Reserved'. - Point is locked.	- Track / Reserved Status of track section containing point is 'Clear' / 'Not Reserved'.		

Desc.	Track section containing locked point is released after Train 2 has run over the point.	
	#1.D	Trains Management receives Train Position Report of Train 2.
	#2.D	Trains Management sends information of track section release by Train 2 to Track Status Management.
	#3.D	Track Status Management sets status of track section to 'Clear'.
	#4.D	Track Status Management reports release of track section to Route Management.
	#5.D	Route Management releases route section containing the point.

Variant	Description	Alternatives	Main case	Impact/Affected Steps
V1	This variant concerns the case of Unknown Track Status.	Points are locked by Track Status Occupied (V1.A) / Unknown (V1.B).	V1.A	
V2	This variants considers the case that Trackside Train Detection is in place.	Train Locations is based on Train Position Reports (V2.A) / Trackside Train Detection (V2.B).	V2.A	
V3	In this variant, the Dispatcher starts an operational procedure.	Dispatcher does not start (V3.A) / does start (V3.B) an operational procedure.	V3.A	V3.B allows point movement in Occupied / Unknown / Reserved Track Area.

Hazards		
ID	Description	Reference/new possible hazard
H-Points-001	A point is moved in an Unknown / Occupied / Reserved Track Status Area with a train over it, or when it is about to pass over it, leading to derailment.	[X2R3 D4.2]

ID	Applicable Operational Rules	Reference
OPE-Generic-1	Where the system permits, the Dispatcher shall, in accordance with non-harmonised rules, remove an area with track status Unknown.	[X2R3 D4.2] [MR D1.1]
OPE-Generic-6	The Dispatcher shall, in accordance with non-harmonised rules, create or extend an Unknown Area flagged as "Sweepable" or "Non-sweepable".	[MR D1.1]
OPE-Generic-7	The Dispatcher shall, in accordance with non-harmonised rules, be able to move a set of points partially or completely located in an Occupied or Unknown Area.	[X2R3 D4.2]
OPE-OS-1	When sweeping an area in ETCS Level 3 On Sight mode, the Driver shall, in accordance with non-harmonised rules, follow operational procedures.	[MR D1.1]
OPE-OS-2	When asked to confirm the line is Clear, the Driver shall, in accordance with non-harmonised rules, observe the track and confirm the status of sections of track joining/diverging from the line over which the train is passing.	[MR D1.1]
OPE-OS-3	When advised by the Driver that a section of line has been examined and observed clear, the operator shall, in accordance	[MR D1.1]

	with non-harmonised rules, clear the status of sections of track joining/diverging from the line over which the train passed where the system allows.	
OPE-OS-4	The operator shall, in accordance with non-harmonised rules, advise the Driver of any specific checks prior to authorising a move in ETCS Level 3 On Sight mode.	[MR D1.1]

ID	Applicable Requirements	Reference
REQ-PTS-1	The L3 Trackside shall prevent movement of points within an Unknown or Occupied Track Status Area, or within a Reserved Status Area, unless using an operational procedure.	[X2R3 D4.2]
REQ-PTS-2	The L3 Trackside shall be configured with Release Points to enable Points to be moved when the area of track containing the Points has Consolidated Track Status Clear and does not contain any part of a Reserved Status Area.	[X2R3 D4.2]
REQ-PTS-3	On request from the TMS, the L3 Trackside shall be able to move points for which all or parts of it is in an area with Track Status Unknown or Occupied, or both.	[X2R3 D4.2]
REQ-PTS-4	When a train is sweeping a set of points, the L3 Trackside shall remove or reduce a Sweepable Unknown Track Status Area from the alternate leg of the points as far as the Fouling Point, in addition to the path that the train takes.	[X2R3 D4.2]

REFERENCES

[X2R3 D4.2] D4.2–Moving Block System Specifications, X2Rail-3 project, Advanced Signalling, Automation and Communication System, Deliverable D4.2, part 1 to 6, 18/12/2020.

[MR D1.1] D1.1 - Report on Moving Block Operational and Engineering Rules, MOVINGRAIL, Deliverable 1.1, section 4.5, 07/12/2020.

Appendix D – OS#4 - Crossing of Radio Hole

Operational Scenario #4	
Title:	Crossing of Radio Hole
Abstract:	In this operational scenario, a connected train moving under the supervision of ETCS L3 enters an active Radio Hole area, in which a blackout in communications is expected.
Description:	The final aim of this operational scenario is to evaluate the time needed by a train to cross a radio hole depending on parameters such as the speed of the train, the quality of the communication network and the radio hole timer. Two cases may occur: in the first one, the parameters are set to values that guarantee the trackside to keep the connection of the train alive during all the disconnection interval; in the second case, the connection with the train is lost and the train does not reconnect to the trackside in a timely manner: in this case a SR exit from the radio hole can be necessary.

Applicable Use Case(s)	
1	Radio Hole
2	Loss/Restore of Communication
3	Normal Train Movement
4	Staff Responsible (SR) movement

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Radio Hole Crossing Time	Quantitative	Performance		Time for the train to cross the radio hole.

Signalling Type	System Type	Track Information
Freight Lines	Full MB (FMB)	

Functional components	
L3 Trackside	
Trackside Train Detection	
ETCS OnBoard	
Train/TIMS	
Driver	
Traffic Management System	

Trackside Function(s)	ETCS On-Board Function(s)
Track Status Management	Train Position Reporting
Reserved Status Management	
Communication Management	

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)	RH_TIMER	To be defined	Radio Hole Timer: assigned by TMS, denotes the expected time the train will take to get through the Radio Hole area.	[X2R3 D4.2]
	TRAIN_COMM_EXPIR	To be defined	Timer of the train after which the connection with the Trackside can be considered terminated	
Train	NID_ENGINE	To be defined	Identifies a train. Used to identify a train when restoring communications with a new session after entering the Radio Hole area.	[X2R3 D4.2]
	L_TRAIN	To be defined	Length of the train.	[X2R3 D4.2]
Speed	V0	To be defined	Initial train speed	
	MAX_SR_SPEED	To be defined	Maximum allowed speed in SR Mode	
Distance	D	To be defined	Distance between the Radio Hole area end and the end of the EoA Exclusion Area.	
Communication	P_COMM	To be defined	Probability of successful communication inside the Radio Hole Area.	
Track	L_RH	To be defined	Length of the radio hole area.	

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	A connected train is authorized to move in FS until the distance D beyond the end of the Radio Hole Area termination point.			
Desc.	L3 Trackside is notified that a connected train is entering a Radio Hole area. L3 Trackside starts the appropriate timer.			

	#1	L3 Trackside receives a PR with the MSFE of the train that passed the initial point of the Radio Hole Area.		
	#2	L3 Trackside stops monitoring the Mute timer, Integrity Wait Timer, ETCS session timer, and activates the Radio Hole timer.		
	#3	The train continues trying to send PRs to the Trackside. According to the probability of communications, the Trackside can receive some of these messages.		
	#4	Position of the train is updated in case of successful messages: track status, RH_TIMER, Reserved status are updated accordingly.		
A		Reserved Status, Track Status and MA are updated accordingly.		
Desc.	The train is fast enough to get through the Radio Hole area before the Radio Hole Timer expires.			
	#5.A	The train exits the Radio Hole Area and continues its march in FS mode before the Radio Hole Timer expires.		
B		The Trackside recognizes the train, and update the Track Status and Reserved Status accordingly to the Position Report.		
Desc.	The train fails to emerge from the Radio Hole area before the Radio Hole Time expires.			
	#5.B	Radio Hole Timer ends, the Track status of the Radio Hole Area is considered Unknown.		
	#6.B	The train loses its communication with the Trackside and starts a braking procedure.		
	#7.B	The train considers the communication with the Trackside terminated after the expiration of TRAIN_COMM_EXPIR.		
	#8.B	The train starts the procedure to move in Staff Responsible		
	#9.B	The train exits the Radio Hole Area and reconnects to the Trackside.		

Variant	Description	Alternatives	Main case	Impact/Affected Steps
V1	This variant is concerned with how train exit from a Radio Hole area is determined.	Train exit from Radio Hole is determined by a PR with (V1.A) MSFE / (V1.B) CSRE outside the Radio Hole Area.	V1.A	#5.A #9.B
V2	This variant explores different approaches in the Radio Hole Timer management.	Radio Hole Timer (V2.A) is / (V2.B) is not reset upon reception of a PR from inside the Radio Hole area.	V2.A	#4

Hazards		
ID	Description	Reference/new possible hazard

--	--	--

ID	Applicable Operational Rules	Reference
[X2R3 D4.2 OPE-LossComms-2]	The Dispatcher shall activate, in accordance with non-harmonised rules, a pre-defined temporary radio hole.	[X2R3 D4.2]
[X2R3 D4.2 OPE-LossComms-3]	The Dispatcher shall deactivate, in accordance with non-harmonised rules, a temporary radio hole.	[X2R3 D4.2]
[X2R3 D4.2 OPE-LossComms-4]	When alerted by the L3 Trackside that a train has been in a radio hole for longer than expected, the Dispatcher shall apply non-harmonised rules.	[X2R3 D4.2]

ID	Applicable Requirements	Reference
[X2R3 D4.2 REQ-RadioHole-1]	L3 Trackside shall activate or deactivate predefined temporary Radio Holes upon request from TMS.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RadioHole-2]	L3 Trackside shall establish an End of Authority Exclusion Area for each Radio Hole.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RadioHole-3]	L3 Trackside shall remove the End of Authority Exclusion Area created for a Temporary Radio Hole if the latter is deactivated.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RadioHole-4]	L3 Trackside shall start the Radio Hole timer when a train enters a Radio Hole.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RadioHole-5]	L3 Trackside shall stop supervising the following timers (Mute timer, Integrity wait timer, ETCS session timer) for a given train when that train is in a Radio Hole area.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RadioHole-6]	Upon expiry of the Radio Hole timer, the L3 Trackside shall behave as in the Loss of Communications use case.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RadioHole-7]	Upon expiry of the Radio Hole timer, the L3 Trackside shall notify the TMS that a train failed to emerge from a Radio Hole.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RadioHole-8]	L3 Trackside shall stop the Radio Hole timer when a train leaves a Radio Hole area.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RecoveryMgmt-1]	L3 Trackside shall consider a train, which starts communicating with the L3 Trackside within the same communications session as previously used for the train, as the same train so long as no change in train data has occurred. This happens when a train leaves a Radio Hole area before the Radio Hole timer expires.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RecoveryMgmt-2]	L3 Trackside shall consider a train reconnecting with a new communication session as the same train of a previous communication session if (a) the two trains have the same ID (NID_ENGINE) and (b) the two trains have the same length (L_TRAIN).	[X2R3 D4.2]
[X2R3 D4.2 REQ-RecoveryMgmt-3]	If the L3 Trackside determines that the same train has reconnected and confirmed Integrity, the L3 Trackside shall update the Unknown Track Status Area associated with this train, resulting from the Loss of Communications due to the expiry of the Radio Hole timer to an Occupied Track Status Area with an extent corresponding to the new train location.	[X2R3 D4.2]

REFERENCES

[X2R3 D4.2] D4.2–Moving Block System Specifications, X2Rail-3 project, Advanced Signalling, Automation and Communication System, deliverable D4.2, part 1 to 6, 18/12/2020.

Appendix E – OS#5 - Loss/Restore of Communications

Operational Scenario #5	
Title:	Loss/Restore of Communications
Abstract:	This scenario analyses the system behaviour in case of loss of communication against known and not considered hazards and/or understanding the sweeping activation conditions.
Description:	In the case of a connected ETCS L3 supervised train, in case that communication with the train is lost, three possible cases can occur: in case A, connection is re-established before session timeout; in case B, connection is re-established before session timeout, with changes in train position/id/length; lastly, in case C, the train fails to re-connect before session timeout.

Applicable Use Case(s)	
1	Loss/Restore of Communication
2	Normal Train Movement
3	Staff Responsible (SR) movement
4	Sweeping
5	Loss of Train Integrity

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Hazard Probability	Quantitative	Safety		Probability of hazard in case of loss/restore of communication
Set of sweeping conditions	Qualitative	Performance		Set of the conditions bringing to sweeping procedure activation

Signalling Type	System Type	Track Information
General	Full MB (FMB)	

Functional components
L3 Trackside
Trackside Train Detection
ETCS OnBoard
Train/TIMS
Driver
Traffic Management System

Trackside Function(s)	ETCS On-Board Function(s)
Track Status Management	Train Position Reporting
Reserved Status Management	
Communication Management	

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)	MUTE_TIMER	To be defined	Defines a timeout for a given train with which L3 Trackside has an active communication session. When this timer expires, the communication with this train is considered lost.	[X2R3 D4.2]
	SESSION_EXPIRED_TIMER	To be defined	Timer of the train after which a communication session is considered terminated.	
Train	NID_ENGINE	To be defined	Identifies a train. Used to identify a train when restoring communications with a new session after entering the Radio Hole area.	[X2R3 D4.2]
	L_TRAIN	To be defined	Length of the train.	[X2R3 D4.2]
Communication	P_COMM	To be defined	Probability of successful communication inside the Radio Hole Area.	

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	A connected train is authorized to move in FS mode.	After loss of communications, Track status for the area in front of the train is updated and set to unknown.		
Desc.	The connected train fails to communicate an updated PR to L3 Trackside before its MUTE_TIMER expires. L3 Trackside considers the communication lost and, hence, takes the necessary safety measures to mitigate risks.			
#1	The connected train fails to send a PR to L3 Trackside before its MUTE_TIMER expires.			
#2	As a result of the expiration of MUTE_TIMER, L3 Trackside considers the communication with the train to be lost.			

	#3	The track area comprised between the latest Confirmed Rear End position of the train and its most recent EoA is considered to have status Unknown.		
A		MA is reassigned to the train, MUTE_TIMER and SESSION_EXPIRE_D_TIMER are reset.		
Desc.	After the loss of communication, the train sends a PR before the SESSION_EXPIRED_TIMER expires, with NID_ENGINE and L_TRAIN being unchanged w.r.t. the last communication. In this case, the communication is restored and track status is cleared.			
	#4.A	L3 Trackside receives a PR from the TRAIN before the SESSION_EXPIRED_TIMER expires, with NID_ENGINE and L_TRAIN being unchanged w.r.t. the last communication.		
	#5.A	Communication is considered restored. The status of the track area comprised between the latest CRE position of the train and its most recent EoA is cleared. Movement authority is reassigned to the train, and both the MUTE_TIMER and the SESSION_EXPIRED_TIMER are reset.		
B				
Desc.	After the loss of communication, the train sends a PR before the SESSION_EXPIRED_TIMER expires, but the train is not recognized as the same train, since either NID_ENGINE or L_TRAIN have changed w.r.t. the last communication. In this case, the communication is restored but the track status is not cleared.			
	#4.B	L3 Trackside receives a PR from the TRAIN before the SESSION_EXPIRED_TIMER expires, but NID_ENGINE and/or L_TRAIN have unchanged w.r.t. the last communication.		
	#5.B	Communication with the train is restored. The portion of unknown track status area is not cleared.		
	#6.B	The unknown track status is cleared by running the train in On Sight mode to sweep the Unknown tracks.		
C				
Desc.	After the loss of communication, the train fails to reconnect before the SESSION_EXPIRED_TIMER expires.			
	#4.C	L3 Trackside does not a PR from the TRAIN before the SESSION_EXPIRED_TIMER expires.		
	#5.C	The train performs the Start of Mission procedure to continue to its destination.		

Variant	Description	Alternatives	Main case	Impact/Affected Steps

Hazards		
ID	Description	Reference/new possible hazard
H-Clearing-001	Track Status Area erroneously cleared during L3 Trackside initialisation by dispatcher leading to a collision. The hazard applicable to this use case is mainly related to the incorrect clearing of tracks after the recovery of communication (after the Mute Timer timeout).	[X2R3 D4.2]

ID	Applicable Operational Rules	Reference
[X2R3 D4.2 OPE-LossComms-1]	The Dispatcher shall, in accordance with non-harmonised rules, protect the movement of a non-communicating train. The movement of a non-communicating train must be safe and controlled by the Driver and the Dispatcher working together.	[X2R3 D4.2]

ID	Applicable Requirements	Reference
[X2R3 D4.2 REQ-LossComms-1]	To timely react to the potential loss of communications with ETCS On-board, the L3 Trackside, if configured to do so, shall be able to supervise a defined timeout (a MUTE_TIMER) for each train with which it has an active communication session. When this timer expires, the communication with this train is considered lost.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossComms-2]	L3 Trackside shall reset the MUTE_TIMER for a train upon receiving of a message from said train.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossComms-3]	The L3 Trackside shall maintain the communication session with ETCS On-board as active even when the MUTE_TIMER has expired until also the maximum time (SESSION_EXPIRED_TIMER) to maintain a communication session has expired. Between the expiry of the MUTE_TIMER and the expiry of the SESSION_EXPIRED_TIMER, the L3 Trackside shall treat the train as having lost communications. However, the connection shall be maintained during this period in case the train regains communications.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossComms-4]	When the Mute timer expires for a train which has not been sent Reversing Area Information, nor entered an announced Radio Hole, then the L3 Trackside shall change the Track Status Area associated with the train to Unknown and extend this Area until the end of the Reserved Status Area for the train.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossComms-5]	If the Mute timer is not considered for use on a particular application, the L3 Trackside shall react when the session timer expires by setting the Track Status Area associated with the train to Unknown and extend this Area until the end of the Reserved Status Area for that train.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossComms-6]	When the L3 Trackside considers the communication session with a train is terminated, then the L3 Trackside shall remove any Reserved Status Area associated with that train.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RecoveryMgmt-1]	L3 Trackside shall consider a train which starts communicating with the L3 Trackside within the same communications session as previously used for the train as the same train, so long as no change in train data has occurred. This happens when a train leaves a Radio Hole area before the Radio Hole timer expires.	[X2R3 D4.2]
[X2R3 D4.2 REQ-RecoveryMgmt-2]	L3 Trackside shall consider a train reconnecting with a new communication session as the same train of a previous communication session if (a) the two trains have the same ID (NID_ENGINE) and (b) the two trains have the same length	[X2R3 D4.2]

	(L_TRAIN).	
[X2R3 D4.2 REQ-RecoveryMgmt-3]	If the L3 Trackside determines that the same train has reconnected and confirmed Integrity, the L3 Trackside shall update the Unknown Track Status Area associated with this train, resulting from the Loss of Communications due to the expiry of the Radio Hole timer to an Occupied Track Status Area with an extent corresponding to the new train location.	[X2R3 D4.2]

REFERENCES

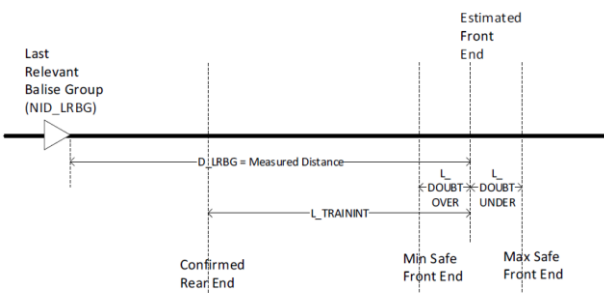
[X2R3 D4.2] D4.2–Moving Block System Specifications, X2Rail-3 project, Advanced Signalling, Automation and Communication System, deliverable D4.2, part 1 to 6, 18/12/2020.

Appendix F – OS#6 - Loss of Train Integrity

Operational Scenario #6	
Title:	Loss of Train Integrity
Abstract:	In this operational Scenario, a connected train moving under the supervision of ETCS L3 train loses its integrity.
Description:	The final aim of this operational scenario is to protect the rear end of the train and other trains from collision in the case that a train has lost its integrity. This may occur for different reasons but in the event that a train splits unintentionally, the Dispatcher needs to take relevant steps to prevent the potentially hazardous situation. Lack of Train Integrity information has a significant impact on the performance of the line.

Applicable Use Case(s)	
1	Normal Train Movement
2	Staff Responsible (SR) movement
3	On Sight(OS) movement
4	Loss/restore of communication

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Loss of integrity duration	Quantitative	Performance		Duration that the train had lost its integrity.
Probability of train integrity loss	Quantitative	Safety		Probability that the train integrity is lost

Signalling Type	System Type	Track Information
Freight Lines	Full MB (FMB)	

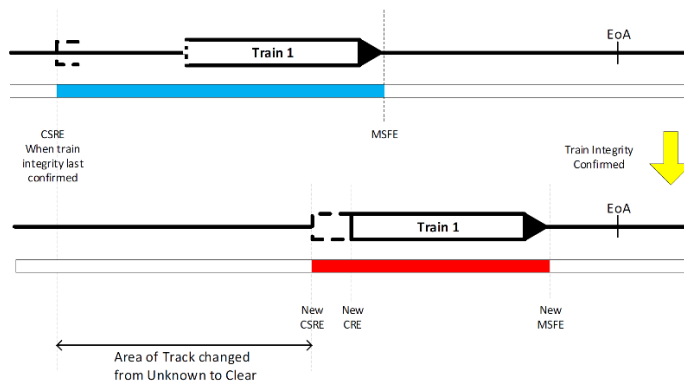
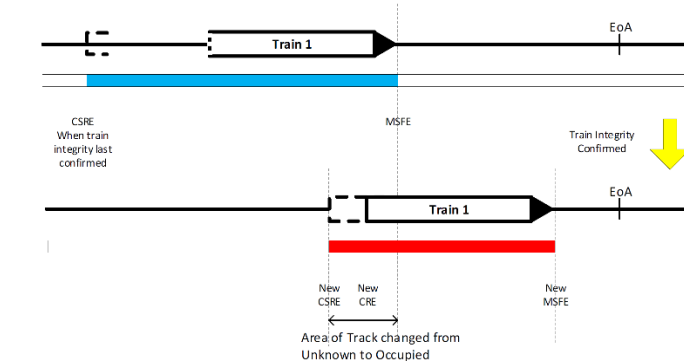
Functional components
L3 Trackside
ETCS OnBoard
Train/TIMS
Dispatcher
Traffic Management System
Trackside Train Detection

Trackside Function(s)	ETCS On-Board Function(s)
Track Status Management	Train Position Reporting
Reserved Status Management	Integrity Information Management
Communication Management	
MA Management	

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)	Train integrity timer	To be defined	timeout related to the maximum time period that the Trackside can wait before considering that the integrity is lost	
Train	NID_LRBG	[0 ... 2 ²⁴]	Identity of last relevant balise group	[SS026 part7]
	D_LRBG	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	Distance between the last relevant balise group and the estimated front end of the train	[SS026 part7]
	Q_LENGTH	0: No TI information available 1: TI confirmed by TIMS [external device] 2: TI confirmed by driver 3: TI lost	Qualifier for train integrity status	[SS026-part7]
	L_DOUBTOVER = MaxSFE	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	over-reading amount (odometry error + error in detection of balises in rear of the estimated train position) + Q_LOCAC (position error in meter) of the LRBG location	[SS026 part7]

	L_DOUBTUNDER = MinSFE	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	under-reading amount (odometry error + error in detection of balises in front of the estimated train position) + Q_LOCACCC (position error in meter) of the LRBG location	[SS026 part7]
	D_LRBG	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	Distance between the last relevant balise group and the estimated front end of the train	[SS026 part7]
	L_TRAIN	[0 ... 4095] Unit: m resolution: 1m	This is the absolute real length of the train.	[SS026-part7]
Speed	MAX_SR_SPEED	To be defined	Maximum allowed speed in SR Mode	
Distance	EoA	To be defined	End of Movement Authority	

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	A connected train is authorized to move in FS MA until an allocated target point EoA.			
Desc.	L3 Trackside receives a position report from a train with the information 'Train integrity lost'.			
	#1	The L3 Trackside updates the established rear end of the train to the assumed rear end		
	#2	<div><p>The L3 Trackside changes the Track Status associated with the train to Unknown (the Track area from the Confirmed Safe Rear End (CSRE) until the Max Safe Front End of the train is changed to Unknown).</p></div>		
	#3	The L3 trackside is not able to extend the FS MA for following train.		
	#4	The TIMS informs the driver of the situation by an indication in the cab		
	#5	The train may continue its mission		
A				

Desc.	the L3 Trackside receives a new message from a train with the information 'Train integrity confirmed by external device TIMS'			
	#6.A	The L3 Trackside shall start/restart the Integrity Wait Timer		
B	L3 Trackside is configured to accept confirmation by Driver			
Desc.	the L3 Trackside receives a new message from a train with the information 'Train integrity confirmed by Driver'			
	#6.B	The L3 Trackside stops the Integrity Wait Timer.		
	#7.B	<p>The Track Status areas between the old CSRE and New CSRE with state Unknown will change to Clear,</p> 		
	#8.B	<p>The Track Status Areas for the train (the state of the areas between the New CSRE and old MSFE) with state Unknown will change to Occupied.</p> 		
	#9.B	the trackside extends the FS MA for following Train.		

Variant	Description	Alternatives	Main case	Impact/Affected Steps
V1	L3 Trackside is not configured to accept confirmation by Driver	The L3 Trackside treats this message as "No train integrity information"		#6.B

Hazards		
ID	Description	Reference/new possible hazard
[X2R3 D4.2 H-Movements-005]	Undetected movement of a part of the train after loss of integrity leading to collision.	[X2R3 D4.2]
[X2R3 D4.2 H-	Undetected movement entering the L3 area	PERFORMINGRAIL WP1

Movements-003]	leading to collision	
[X2R3 D4.2 H-Clearing-003]	Track Status Area erroneously cleared after deactivation of a shunting area leading to collision.	PERFORMINGRAIL WP1
[X2R3 D4.2 H-TTDfailure-001]	TTD erroneously indicates a Clear Track Status Area leading to collision or derailment	PERFORMINGRAIL WP1
[X2R3 D4.2 H-Level2-003]	Derailment after loss of train integrity causes adjacent tracks to become occupied leading to collision.	[X2R3 D4.2]

ID	Applicable Operational Rules	Reference
[X2R3 D4.2 OPE-LossTI-2]	When advised of loss of train integrity, the Dispatcher shall, in accordance with non-harmonised rules, protect the area in which a train division may have occurred.	[X2R3 D4.2]
[X2R3 D4.2 OPE-LossTI-2]	When advised of loss of train integrity through an in-cab indication, the Driver shall follow non-harmonised rules	[X2R3 D4.2]
[X2R3 D4.2 OPE-LevelTrans-2]	When TIMS is not working or the train is not reporting train integrity confirmed and the Level 3 trackside is engineered not to authorise such trains to enter, the Dispatcher shall apply non-harmonised rules whether to authorise a train to enter a Level 3 Only area.	[X2R3 D4.2]
[X2R3 D4.2 OPE-Generic-2]	The Driver shall only confirm train integrity in accordance with non-harmonised Operational Rules.	[X2R3 D4.2]
[X2R3 D4.2 OPE-StartTrain-2]	Non-harmonised Operational Rules shall define under which circumstances the Driver is allowed to move a train which is not able to report integrity confirmed.	[X2R3 D4.2]

ID	Applicable Requirements	Reference
[X2R3 D4.2 REQ-LossTI-1]	When receiving a position report from a train with the information 'Train integrity lost', the L3 Trackside shall change the Track Status Area associated with this train to Unknown.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossTI-2]	When the L3 Trackside considers that the integrity is lost for a train, the L3 Trackside shall change the Track Status Area associated with this train to Unknown.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossTI-3]	When the L3 Trackside considers that the Train Integrity is lost for a train, the L3 Trackside shall react as configured.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossTI-4]	The L3 Trackside shall consider the Train Integrity as lost when 'No train integrity information' is reported longer than a configurable time (Integrity Wait Timer).	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossTI-5]	When receiving a message from a train with the information 'Train integrity confirmed by external device', the L3 Trackside shall start/restart the Integrity Wait Timer.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossTI-6]	When receiving a message from a train with the information 'Train integrity confirmed by Driver', if the L3 Trackside is configured to accept confirmation by Driver and the Integrity Wait Timer is running, then the L3 Trackside shall stop the Integrity Wait Timer.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossTI-7]	After a loss of integrity, the driver shall be made aware of the situation via an indication in the cab.	[X2R3 D4.2]

[X2R3 D4.2 REQ-LossTI-8]	The L3 Trackside shall be able to be configured whether to accept Train Integrity confirmation by the driver.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossTI-9]	The L3 Trackside shall be configurable as to whether it authorises a Movement Authority for a train that has lost Integrity.	[X2R3 D4.2]
[X2R3 D4.2 REQ-LossTI-10]	If the L3 Trackside receives Validated Train Data for a train with a train length different from previously reported within the same communication session, then the L3 Trackside shall consider the train as having lost Integrity.	[X2R3 D4.2]
[X2R1 D5.1 REQ-LossTI-1]	When receiving loss of train integrity information from the ETCS On-board, the L3 Trackside shall maintain the Confirmed Rear End (CRE) at the last known location at which the train reported Integrity Confirmed, and the associated Confirmed Safe Rear End (CSRE).	[X2R1 D5.1]
[X2R1 D5.1 REQ-LossTI-2]	At the time that loss of Integrity is reported, the L3 Trackside shall consider as Unknown the area from the Confirmed Safe Rear End (CSRE) until the Max Safe Front End of the train.	[X2R1 D5.1]
[X2R1 D5.1 REQ-LossTI-3]	For a train that has never confirmed integrity, the L3 Trackside shall consider as Unknown the area from the last rear end boundary until the Max Safe Front End of the train.	[X2R1 D5.1]
[X2R1 D5.1 REQ-LossTI-6]	If, after reporting loss of train integrity, the ETCS On-board reports integrity confirmed again, the L3 Trackside shall change the state of the areas between the old CSRE and New CSRE with state Unknown to Clear if: the L3 Trackside is able to locate the train unambiguously, and <ul style="list-style-type: none"> no other obstacle has entered the Unknown area since train integrity was lost. 	[X2R1 D5.1]
[X2R1 D5.1 REQ-LossTI-7]	If, after reporting loss of train integrity, the ETCS On-board reports integrity confirmed again, the L3 Trackside shall change the state of the areas between the New CSRE and old MSFE with state Unknown to Occupied if <ul style="list-style-type: none"> the L3 Trackside is able to locate the train unambiguously, and no other obstacle has entered the Unknown area since train integrity was lost. 	[X2R1 D5.1]

REFERENCES

[X2R3 D4.2] D4.2–Moving Block System Specifications, X2Rail-3 project, Advanced Signalling, Automation and Communication System, deliverable D4.2, part 1 to 6, 18/12/2020.

[X2R1 D5.1] D5.1–Moving Block System Specifications, X2Rail-1 project, 01/09/2016.

Appendix G – OS#7 - Shunting Movement

Operational Scenario #7	
Title:	Shunting Movement
Abstract:	ETCS includes a mode called shunting (SH), which enables trains to be moved both forwards and backwards and without the need for the trackside to issue movement authorities. Having granted permission for the train to enter SH, the trackside has very restricted functionality available to manage the train movement or to restrict it from entering an operational line leading to collision.
Description:	This operational scenario assumes that the ETCS Level 3 moving block is able to manage a possible driver's request for shunting anywhere on the line, but could decide to reject this and restrict shunting to predefined shunting areas. We describe two variants, one that considers the train entering the temporary shunting area manually, and the other one entering the same area automatically.

Applicable Use Case(s)	
1	Shunting Train Movement
2	Normal Train Movement
3	Loss/Restore of Communication
4	End of Mission

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Average time to resume normal driving	Quantitative	Performance		Average time for the train to cross the shunting area
Probability of unauthorized exit from shunting area	Quantitative	Safety	Probability of dangerous failure per hour (PFH): $10^{-8} - 10^{-9}$	


Signalling Type	System Type	Track Information
General	Full MB (FMB)	Predefined areas in the track dedicated to shunting/Predefined shunting areas that can be activated/deactivated under Traffic Management System control

Functional components
ETCS OnBoard
Adjacent Signalling System
Traffic Management System
Balises
Train electromechanical components
Driver

Trackside Function(s)	ETCS On-Board Function(s)
L3 Trackside	Train Position Reporting
Temporary Shunting Area Management	Speed Supervision

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Train	L_TRAIN	To be defined	Length of the train	[X2R3 D4.2]
Track (Shunting Area)	L_SH	To be defined	Length of shunting area	[X2R3 D4.2]
	Type_SH	Temporary/Permanent	Type of shunting	[X2R3 D4.2]
Speed	V0	To be defined	Train speed when entering the shunting area	[X2R3 D4.2]
	MAX_SH_SPEED	To be defined	Maximum allowed speed in SH mode	[X2R3 D4.2]
Balise Communication	B_COMM	To be defined	Probability of successful balise read within the shunting area	[ERA_ERTMS_OP E_App A_5 V 5.0]

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	The Traffic Management System grants the SH request. The train is stationary and fully inside the inactive temporary shunting area.		Driver requests for shunting.	
Desc.	The L3 trackside activates a temporary shunting area.			
	#1	L3 trackside sets the track status within the active shunting area to unknown.		
A				
Desc.	Manual entry into shunting			
	#2.A	The driver selects "Shunting" and the train enters the shunting area and performs forward/backward shunting movements within SH area.		
	#3.A	The ECTS on-board communicates its position with the balise.		
	#4.A	The train reaches the border of the shunting area and the driver selects "exit shunting", ensuring that no traction unit remains in the "Maintain Shunting" status.		
	#5.A	L3 trackside receives "end of shunting" message and removes the track status area associated to the train.		
B	The following symbol is displayed with a			

	flashing frame: 			
Desc.	Automatic entry into shunting			
	#2.B	The train receives information from the L3 trackside and acknowledges. The train enters the shunting area and performs forward/backward shunting movements within SH area.		
	#3.B	The ECTS on-board communicates its position with the balise.		
	#4.B	The train reaches the border of the shunting area and the driver selects “exit shunting”, ensuring that no traction unit remains in the “Maintain Shunting” status.		
	#5.B	L3 trackside receives “end of shunting” message and removes the track status area associated to the train.		
	The Traffic Management System does not grant the SH request.		Driver requests for shunting.	
Desc.	The L3 trackside rejects the shunting request and does not activate the temporary SH area.			
	#6	L3 trackside send the message “SH refused” to the train.		
	#7	Driver applies non-harmonised rules.		

Variant	Description	Alternatives	Main case	Impact/Affected Steps
V1	This variant is concerned with temporary shunting areas that are entered manually.	On request from the Traffic Management System, the L3 Trackside activates a temporary shunting area, which is entered manually by the train.	V1.A	#2.A
V2	This variant explores the situation of temporary shunting areas entered automatically.	The L3 trackside is configured with a temporary shunting area which is entered automatically by the train.	V1.B	#2.B

Hazards		
ID	Description	Reference/new possible hazard
#1: Undetected movement out of an activated shunting area leading to collision	Shunting movements may unintentionally move beyond the border of an active shunting area without the L3 Trackside being aware of this and therefore being unable to protect other movements in the vicinity of the shunting area.	[X2R3 D4.2]
#2: Track Status Area erroneously cleared after deactivation of a shunting area leading to	The L3 Trackside considers the track status in an active shunting area as Unknown Track Status Area, except for the location of communicating trains. When deactivating a shunting area, responsible staff may have the possibility to clear any remaining Unknown Track Status Area. Doing this, an Occupied Track Status Area can be set to	[X2R3 D4.2]

collision	Clear leading to collision.	
-----------	-----------------------------	--

ID	Applicable Operational Rules	Reference
[MOVINGRAIL D1.1 OPE-SH-1]	The operator shall, in accordance with non-harmonized rules, activate temporary shunting areas, where needed.	[MOVINGRAIL-D1.1]
[MOVINGRAIL D1.1 OPE-SH-2]	The operator shall, in accordance with non-harmonized rules, deactivate temporary shunting areas.	[MOVINGRAIL-D1.1]
[MOVINGRAIL D1.1 OPE-SH-3]	The operator shall, in accordance with non-harmonized rules, allow trains to enter a shunting area.	[MOVINGRAIL-D1.1]

ID	Applicable Requirements	Reference
[X2R3 D4.2 REQ-SH-1]	The L3 Trackside shall be configurable with predefined Permanent and Temporary Shunting Areas.	[X2R3 D4.2]
[X2R3 D4.2 REQ-SH-2]	On request from the TMS, the L3 Trackside shall be able to activate and deactivate a Temporary Shunting Area.	[X2R3 D4.2]
[X2R3 D4.2 REQ-SH-3]	The L3 Trackside shall consider the Track Status of an Active Shunting Area to be Unknown and Non-Sweepable.	[X2R3 D4.2]
[X2R3 D4.2 REQ-SH-4]	The L3 Trackside shall perform specific checks before activating a Temporary Shunting Area.	[X2R3 D4.2]

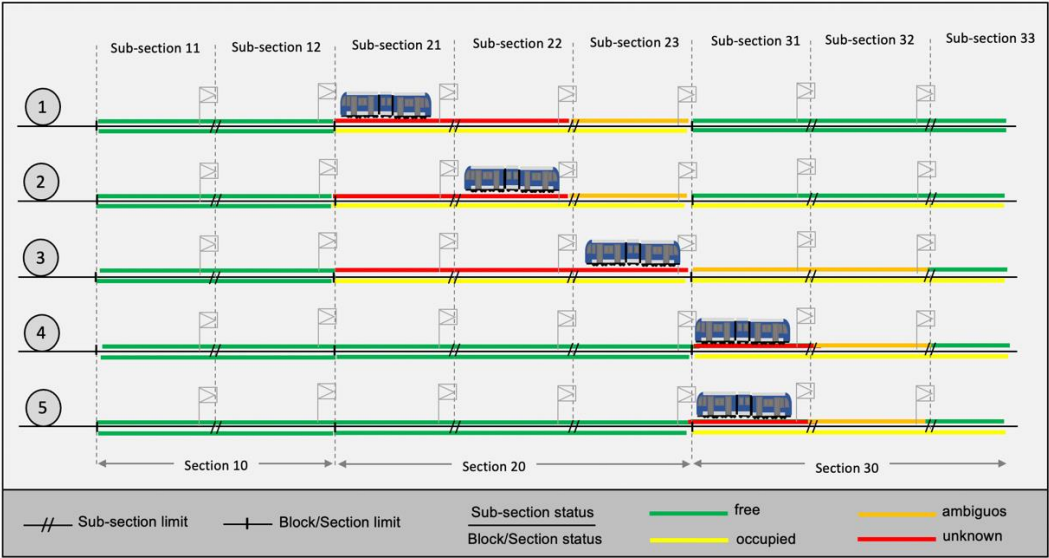
REFERENCES

[X2R3 D4.2] D4.2–Moving Block System Specifications, X2Rail-3 project, Advanced Signalling, Automation and Communication System, deliverable D4.2, part 1 to 6, 18/12/2020.

[MOVINGRAIL-D1.1] Deliverable D1.1 - Report on Moving Block Operational and Engineering Rules, 07/12/2020.

[ERA_ERTMS_OPE_App A_5 V 5.0] ERTMS Operational Principles and Rules ERA_ERTMS_OPE_App A_5 / V 5.0. European Union Agency for Railways, 2013.

Appendix H – OS#8 - End of Mission

Operational Scenario #8	
Title:	End of Mission
Abstract:	This scenario describes the The End of Mission (EoM) process for an L3 Area.
Description:	<p>General context of SoM:</p> <p>When a train completes a journey and the Driver closes the desk, the onboard issues an EoM request and the train disconnects.</p> <p>Based on [ERTMS/ETCS L3 principles], the end of mission scenario is represented in Figure 1:</p>  <p>The steps to consider in this scenario are:</p> <p>Step 1 - Train 2 moves to sub-section 21 which becomes “occupied” and all sub-sections in block 10 become “free”.</p> <p>Step 2 - Train 2 continues to sub-section 22 which becomes “occupied”.</p> <p>Step 3 - Train 2 performs the End of Mission procedure.</p> <p>Step 4 - Due to the EoM procedure sub-section 22 goes to “unknown” and the disconnect propagation timer is started.</p> <p>Step 5 - The disconnect propagation timer of sub-section 22 expires. All remaining sub-sections in block 20 go to “unknown”.</p>

Applicable Use Case(s)	
1	End of Mission

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description

Completeness	Quantitative	Reliability, Safety	10e-2	The probability that an object-status remains unknown (null)
Reliability	Quantitative	Reliability	10e-9	The probability that a vital object state value is wrong

Signalling Type	System Type	Track Information
General	Full MB (FMB)	

Functional components
L3 Trackside
ETCS On-board
Traffic Management System
Train/TIMS
Driver

Trackside Function(s)	ETCS On-Board Function(s)
Communication Management	Train Position Reporting
Trains Management	Integrity Information Management
Track Status Management	Speed and Distance Supervision

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)	disconnect propagation timer	[5 ... 15min]	Time related to no response from the train	
Train	D_LRBG	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	Distance between the last relevant balise group and the estimated front end of the train	[SS026 part7]
	Q_LOCACC	[0 ... 63] unit: m resolution: 1m	Accuracy of the balise location.	[SS026 part7]
	Q_NVLOCACC	[0 ... 63] unit: m resolution: 1m	Default accuracy of the balise location (absolute value).	[SS026 part8]
	L_DOUBTUNDER	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10	under-reading amount (odometry error + error in detection of balises in	[SS026 part7]

		m (according to Q_SCALE)	front of the estimated train position) + Q_LOCACC (position error in meter) of the LRBG location	
	L_DOUBTOVER:	[0 ... 327 660] unit: km resolution: 10 cm, 1m or 10 m (according to Q_SCALE)	The over-reading amount plus the Q_LOCACC of the LRBG.	[SS026 part7]
	L_TRAIN	[0 ... 4095] Unit: m resolution: 1m	This is the absolute real length of the train.	[SS026 part7] [SS026 part8]

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	<ul style="list-style-type: none"> - The level stored is L3, Train Running Number and RBC data are valid. - The train has already performed the start of mission procedures (i.e., Session with the trackside is established and integrity is confirmed.) - The train has itinerary assigned in execution. 			
Desc.	When a train completes a journey and the Driver closes the desk, the onboard issues an EoM request and the train disconnects. Since after an EoM a train is no longer authorised to move, all Reserved Status Area associated with the train has to be removed.			
	#1	Onboard issues an End of Mission request		
	#2	Track status management removes every Reserved Status Area associated with the train.		
A				
Desc.	<p>After receiving an EoM, The L3 Trackside needs to determine the area that could be occupied by the train to protect it.</p> <p>Assumption 1: The L3 Trackside is able to cope with differences in the confidence of the interval provided in the report transmitted at the EoM, due to an ambiguity in the ETCS specifications around how to calculate the train location accuracy when linking information is deleted due to the change to SB mode.</p> <p>Assumption 2: the L3 Trackside does not apply a safe reaction if a train reports a different</p>			

	confidence interval without the train moving (i.e., different L_DOUBTOVER and L_DOUBTUNDER with the same D_LRBG and LRBG).		
	#3.A	Track status management uses the location information received from the train to determine the area on which the train is located.	
	#4.A	Track Status Management sets the status of the section on which the train is section to 'unknown'.	
B			
Desc.	<p>To avoid unwanted train movement after the disconnection occurs, the state “unknown” shall be propagated, as soon as the disconnect propagation timer expires, onto adjacent sub-sections, forward and backward, until one of the following sections is reached: (i) Free TTD; (ii) Sub-section with a connected train</p> <p>Assumption: The value of the disconnect propagation timer varies between 5-15min [ERTMS/ETCS L3 principles].</p>		
	#3.B	Track status management propagates the “unknown” state onto adjacent subsections.	

Variant	Description	Alternatives	Main case	Impact/Affected Steps

Hazards		
ID	Description	Reference/new possible hazard
H-TrainLoc-001	<p>Error in Train Location from reduced confidence interval at EoM leads to collision.</p> <p>Lack of Train Integrity information, in this case, has a significant impact on the performance of the line. Therefore, it is important that the L3 Trackside receives a recent Train Position Report with the Integrity Confirmed just before EoM. If this does not happen, then there is the potential for a large area of the railway remaining unavailable, e.g., when the CRE remains over points and crossings.</p>	[D4.2–X2Rail-3 2020]

ID	Applicable Operational Rules	Reference
ENG-EoM-1	A margin shall be engineered in the L3 Trackside to establish safely the location of a train that has disconnected after an End of Mission procedure.	[MR D1.1: REQ-EoM-1]
ENG-EoM-2	Infrastructure Manager shall consider the provision of TTD in areas where trains are regularly left without a communication session. Whether a TTD is needed for the vicinity depends on the used technology (axle counters vs track circuits).	[MR D1.1: REQ-EoM-2]

ID	Applicable Requirements	Reference
REQ-EoM-1	The L3 Trackside shall update the stored information of the train performing the EoM.	[X2R1 D5.1: REQ-EoM-1]

REQ-EoM-2	When receiving an End of Mission message from a train which is completely located inside an Active Shunting Area, then the L3 Trackside shall remove the Track Status Area associated with this train.	[X2R3 D4.2-Part3: new]
REQ-EoM-3	When the L3 Trackside receives an End of Mission message, then the L3 Trackside shall remove any associated Reserved Status Area associated with this train.	[X2R1 D5.1: REQ-EoM-5]
REQ-EoM-4	The L3 Trackside shall be able to cope with differences in the confidence interval provided in the position report of a train that reported End of Mission even when related to the same train position.	[X2R1 D5.1: REQ-EoM-6]

REFERENCES

[D5.1–X2Rail1 2019] D5.1–Moving Block System Specification. X2Rail-1 project: “Start-up activities for Advanced Signalling and Automation Systems”, deliverable D5.2, 126 pages, 2019.

[D4.2–X2Rail-3 2020] D4.2–Moving Block System Specifications, X2Rail-3 project, Advanced Signalling, Automation and Communication System, deliverable D4.2 Part 1 to 6, 2020.

[D1.1-MovingRail 2020] D1.1-Report on Moving Block Operational and Engineering Rules, MovingRail project, deliverable D1.1, 41 pages, 2020.

[ERTMS/ETCS L3 principles] Principles of Hybrid ERTMS/ETCS Level 3, EEIG ERTMS Users Group, version 1A, 48 pages, 2017.

[SS026 part 7] ERTMS/ETCS Subset 026–System Requirements Specification, Chapter 7: ERTMS/ETCS language, Issue 3.6.0, 13/05/2016.

[SS026 part8] ERTMS/ETCS Subset 026–System Requirements Specification, Chapter 8: Messages, Issue 3.6.0, 13/05/2016.

Appendix I – OS#9 - Supervising Distance in Normal VCTS Driving

Operational Scenario #9	
Title:	Supervising Distance in Normal VCTS Driving
Abstract:	Supervision of train separation of a Virtual Coupled Train Set (VCTS) during normal driving.
Description:	<p>This operational scenario addresses the supervision of train separation distance during normal driving in Virtual Coupling, and specifically, it assumes that Virtual Coupling has been already initiated.</p> <p>The scenario starts with a VCTS (made of at least two trains) running under nominal Virtual Coupling conditions and aims at evaluating VCTS system safety and performance.</p>

Applicable Use Case(s)	
1	Virtual Coupling – Supervising Train Separation Distance during normal driving

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Probability of hazards due to positioning or communication faults	Quantitative	Safety	SIL4	Probability of having an incorrect safe distance due to positioning errors or delay/errors/loss of communication
Line capacity	Quantitative	Performance		Measuring the expected increase of the line capacity compared to non-VCTS

Signalling Type	System Type	Track Information
General	Full MB (FMB)	Plain track

Functional components
ETCS OnBoard
L3 Trackside
Train/TIMS
VCTS

Trackside Function(s)	ETCS On-Board Function(s)
MA Management	Train Position Reporting
Trains Management	VCTS functions
	Integrity Information Management

Parameters				
	Name	Value/Range	Description (if	Reference

			needed)	(Standards, Deliverables, etc.)
Train		To be defined	Physical characteristics of the trains (length, mass, braking and acceleration/deceleration capacity, maximum speed, etc.)	
Speed	V_0	To be defined	Initial speed	
	a_0	To be defined	Initial acceleration	
Distance		To be defined	Headways of the trains	
Communication		To be defined	Performance and reliability of the communication facilities (train-to-train and train-to-trackside), such as throughput, latency, bit error rate, etc.	
Track		To be defined	Physical characteristic of the track	
Position		To be defined	Positioning precision of the trains due to satellite positioning and/or odometrical error	

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	<p>A master and a slave trains are already virtually coupled</p> <p>The platoon is running (not necessarily at the same speed). Positioning errors may also apply and differences in speed and accuracy may apply.</p> <p>Slave monitor relative distance from master train and from adjacent slaves (if</p>			

	any). The relative distance takes into consideration safety factors due to (list not exhaustive) difference in relative speed and acceleration, latency of the communication channel, positioning error, and different braking capabilities.			
Desc.				
	#1	Master and slave are running along a line, keeping a proper and safe relative distance		
A				
Desc.	VCTS acceleration			
	#2.A	The master accelerates		
	#2.B	The slave accelerates to keep the same safe distance from master		
B				
Desc.	VCTS deceleration/braking			
	#2.A	The master decelerates/brakes		
	#2.B	The slave decelerates/brakes to keep the same safe distance from master		
C				
Desc.	Loss of Communication			
	#2.A	The slave brakes to safely increase the distance from the preceding train		

Variant	Description	Alternatives	Main case	Impact/Affected Steps
V1	Presence vs absence of more than 2 coupled trains			
V2	Presence vs absence of train-to-train communication			

Hazards		
ID	Description	Reference/new possible hazard
	<p>Incorrect safe distance due to positioning errors (odometer and/or satellite)</p> <p>Delay in update of safe distance due to burst communication errors or loss of connection with master or trackside</p>	

ID	Applicable Operational Rules	Reference

ID	Applicable Requirements	Reference

REFERENCES

[X2R3 D6.1] D6.1– Initiation of Virtual Coupling, X2Rail-3 project, Advanced Signalling, Automation and Communication System, Deliverable D6.1

Appendix J – OS#10 - Splitting of a VCTS Initiated by Slave

Operational Scenario #10	
Title:	Splitting of a VCTS Initiated by Slave
Abstract:	Termination of a Virtual Coupling session by splitting of a VCTS initiated by a slave.
Description:	<p>This operational scenario addresses the termination of a Virtual Coupling session, and in particular, the splitting of a VCTS initiated by a slave.</p> <p>The scenario starts with a VCTS running under normal Virtual Coupling driving and ends with two VCTSs (possibly two standalone trains) running under Moving Block signalling.</p>

Applicable Use Case(s)	
1	Virtual Coupling – Splitting of a VCTS Initiated by Slave
2	Virtual Coupling – Supervising Train Separation Distance during normal driving
3	Moving Block – Normal Train Movement

Performance Indicators				
Name	Type	Property (i.e., Logical, Functional, Availability, Reliability, Safety, Performance)	Threshold/Range (if applicable)	Description
Probability of collision	Quantitative	Safety	SIL4	Probability that the relative distance between two trains in the VCTS becomes zero or less
Splitting time	Quantitative	Performance		The time it takes for a slave train to split from a VCTS

Signalling Type	System Type	Track Information
Virtual Coupling	Full MB (FMB)	Plain track

Functional components
L3 Trackside
ETCS OnBoard
Train/TIMS
VCTS

Trackside Function(s)	ETCS On-Board Function(s)
MA Management	Train Position Reporting
Trains Management	VCTS functions
	Integrity Information Management
	Manage Dynamic Speed Profile
	Speed and Distance Supervision

Parameters				
	Name	Value/Range	Description (if needed)	Reference (Standards, Deliverables, etc.)
Timer(s)		To be defined		
Train		To be defined		
Speed	Speed	To be defined		
	Relative speed	To be defined	Speed difference between a slave train and the master train (or preceding train)	
Distance	Distance	To be defined		
	Relative distance	Exceeding the safety margin SM to be defined	Distance gap between a slave train to its predecessor	
Communication		To be defined		
Track		To be defined		
Position		To be defined		

Behaviour				
Branch	Pre-conditions	Post-conditions	Trigger	Invariants/Assertions/...
	VCTS of two trains with relative distance shorter than absolute braking distance	VCTS of two trains with relative distance exceeding absolute braking distance		
Desc.	Slave increases train separation to master to absolute braking distance			
	#1	Slave slows down (coasting or braking)		
	#2	Train separation exceeds absolute braking distance		
	Slave follows master at more than absolute braking distance	Slave train receives moving block MA		Note: order of this step and the next is still unclear
Desc.	Old slave train starts L3 mission			
	#3	Old slave train sends Start of Mission request to L3 trackside Communications Management		
	#4	Old slave train sends Train Position Report to Trains Management		
	#5	Movement Authority Management receives MA request from Trains Management		
	#6	Movement Authority Management creates MA to rear of old master train		
	#7	Movement Authority Management sends MA to ECTS On-Board of old slave train		
	Slave train has its own moving block MA	VCTS split in two separate trains running under moving block		
Desc.	Termination of virtual coupling session			

#8	Slave sends termination of virtual coupling session to master train
#9	Old master train removes slave from cooperative braking

Variant	Description	Alternatives	Main case	Impact/Affected Steps
V1	VCTS of 4 trains splitting in two VCTS of 2 trains each	Number of trains in the VCTS	Last two slaves become a new VCTS with the new leading train taking over the role of the master. Procedure for this slave to become the master to the following slave units needs to be added.	V1

Hazards		
ID	Description	Reference/new possible hazard
	The (old) master applies full braking immediately after termination of the VC session when the cooperative braking of the master train no longer considers the split trains.	X2RAIL-3 D6.1

ID	Applicable Operational Rules	Reference

ID	Applicable Requirements	Reference
	VCTS requirements are in a confidential deliverable from X2RAIL-3	X2RAIL-3 D7.2

Appendix K – Survey questions

PERFORMINGRAIL: survey for the selection of operational scenarios

Among its objectives, PERFORMINGRAIL (<https://www.performingrail.com/>) aims to define and develop formal and semi-formal models of Moving Block (MB) and Virtual Coupling (VC) signalling to validate specifications and perform qualitative and quantitative analysis to support Verification & Validation processes towards a fail-safe and effective signalling configuration.

We defined the following 10 different operational scenarios (i.e., concrete sequences of actions/events of the ETCS entities and external actors in a specific railway configuration) from the available documents, with the goal of evaluating some indices of interest (e.g., performance, availability, safety):

- OS#1 - Trackside initialisation
- OS#2 - Start of Mission
- OS#3 - Points Control
- OS#4 - Crossing of Radio Hole
- OS#5 - Loss/Restore of Communications
- OS#6 - Loss of Train Integrity
- OS#7 - Shunting Movement
- OS#8 - End of Mission
- OS#9 - Supervising Distance in Normal VCTS Driving
- OS#10 - Splitting of a VCTS Initiated by Slave

For each operational scenario, we reported the abstract, a short description, and the set of performance indicators (i.e., the final objective of the evaluation) we have identified so far. This survey aims to receive your comments and feedback and will help us to select the first set on which to focus our attention.

For each scenario, you will be asked to express your evaluation according to the following criteria:

- Significance for market segments/signalling systems
- Safety challenges
- Industrial relevance

We would be very grateful if you could also provide any other suggestions or indications by the open-ended fields. We would like to thank you in advance for the time you will invest in filling the survey!

IMPORTANT NOTE: Please note that the survey is anonymous; no personal data are collected.

1) Operational Scenario #1 - Trackside initialisation

Abstract:

This scenario describes the process of initialising the trackside control systems with up-to-date

values.

Description:

The concept of state vector and its initialisation is central to this scenario. Trackside in this context is the area under control of an RBC or more general, of a Central Safety System that combines RBC and interlocking.

Trackside area is an area that can be located, both geographically and in terms of track topology. State-space represents the status of the trackside system. This is represented by a state vector, i.e. a vector of (object, state)-tuples. Objects include fixed trackside elements such as points as well as transient objects such as trains and temporary speed restrictions.

State vector initialisation allocates state to the values. State is acquired through sensors, actuators (en-UK: detected), and messages.

Performance indicators:

- Average startup time - time the system needs to reach operational status (availability, quantitative);
- Completeness - probability that an object-status remains unknown (reliability & safety, quantitative);
- Safety - probability that a vital object state value is detected wrongly (safety, quantitative).

1.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

1.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

2) Operational Scenario #2 - Start of Mission

Abstract:

This scenario concerns the Start of Mission (SoM) of a non-localised train followed by Staff Responsible to re-locate the train. When the driver begins the Start Of Mission procedure, the train Position Report (PR) status is "Unknown" and L3 Trackside authorizes the train to run in Staff Responsible mode until the train reaches a first location reference and reports its position to the L3 Trackside.

Description:

A stopped train in Stand-by mode and under the supervision of the L3 system has to start a mission with a SoM procedure to reach Full Supervision, On-Sight, Limited Supervision, or Staff Responsible modes. When the SoM procedure is launched, the train cab desk is assumed to be

already open by the driver and, no communication session is still established or being established between the on-board and trackside parts.

The final aim of this operational scenario is to allow the analysis of the success/failure to obtain a first correct and valid position after the SoM procedure. This can be especially analysed when the SoM procedure is started somewhere on a line only equipped with virtual balises. Virtual Block (VB) hazards are related to GNSS-based VB reader issues, in particular, issues related to GNSS feared events.

Performance indicators:

- Probability for the first position to be erroneous while L3 trackside receives a valid PR (i.e. format correct but real position wrongly bounded) (safety, quantitative).

2.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

2.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

3) Operational Scenario #3 - Points Control

Abstract:

This scenario concerns the moving, locking, and releasing of points related to two subsequent trains requiring different points position passing over.

Description:

In this scenario, the situation is considered in which two trains running under normal moving block conditions cross a point consecutively, with the second train requiring the point to move to a different position.

This point cannot be moved as long as the first train is occupying the associated track area. Also, the point movement cannot be done when the point is already reserved for the second train.

These point movement timing restrictions apply due to the hazard of moving a point while a train is passing, or about to pass, over it, possibly leading to derailment of the train.

Performance indicators:

- Headway time - minimum time between train heads over points (performance, quantitative).

3.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
--	-----------	------	--------	-----	----------

Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

3.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

4) Operational Scenario #4 - Crossing of Radio Hole

Abstract:

A connected train moving under the supervision of ETCS L3 enters an active Radio Hole area, in which a blackout in communications is expected.

Description:

The final aim of this operational scenario is to evaluate the time needed by a train to cross a radio hole depending on parameters as the speed of the train, the quality of the communication network, and the radio hole timer. Two cases can occur: in the first, the parameters are set to values that guarantee the trackside to keep alive the connection of the train during all the disconnection interval; in the second case, the connection with the train is lost and the train is not so fast to reconnect to the trackside in a timely manner: in this case, an SR exit from the radio hole can be necessary.

Performance indicators:

- *Radio Hole average crossing time - average time for the train to cross the radio hole (performance, quantitative).*

4.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

4.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

5) Operational Scenario #5 - Loss/Restore of Communications

Abstract:

This scenario analyses the system behaviour in case of loss of communication against known and not known hazards and/or understanding the sweeping activation conditions.

Description:

In the case of a connected ETCS L3 supervised train, if communication with the train is lost, three possible cases can occur:

(A) the connection is re-established before session timeout;

(B) the connection is re-established before session timeout, with changes in train position/id/length;

(C) the train fails to re-connect before session timeout.

Performance indicators:

- *Hazard Probability - probability of hazard in case of loss/restore of communication (safety, quantitative);*
- *Set of sweeping conditions - set of conditions bringing to sweeping procedure activation (performance, qualitative).*

5.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

5.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

6) Operational Scenario #6 - Loss of Train Integrity

Abstract:

In this operational scenario, a connected train moving under the supervision of ETCS L3 train loses its integrity.

Description:

The final aim of this operational scenario is to protect the rear end of the train and other trains from collisions in the case that a train has lost its integrity. It is possible that occurs, for different reasons but in the event that a train is unintentionally divided, the Dispatcher needs to take relevant steps to prevent the potentially hazardous situation. Lack of Train Integrity information has a significant impact on the performance of the line.

Performance indicators:

- *Loss of integrity duration - duration that the train had lost its integrity (performance, quantitative);*
- *Probability of train integrity loss - probability that the train integrity is lost (safety, quantitative).*

6.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

6.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

7) Operational Scenario #7 - Shunting Movement

Abstract:

ETCS includes a mode called shunting (SH), which enables trains to be moved both forwards and backwards and without the need for the trackside to issue movement authorities. Having granted permission for the train to enter SH, the trackside has very restricted functionality available to manage the train movement or to restrict it from entering an operational line leading to collision.

Description:

This operational scenario assumes that the ETCS Level 3 moving block is able to manage a possible driver's request for shunting anywhere on the line, but could decide to reject this and restrict shunting to predefined shunting areas. We describe two variants, one that considers the train entering the temporary shunting area manually, and the other one entering the same area automatically.

Performance indicators:

- Average time to resume normal driving - average time for the train to cross the shunting area (performance, quantitative);
- Probability of unauthorized exit from shunting area (safety, quantitative).

7.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

7.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

8) Operational Scenario #8 - End of Mission

Abstract:

This scenario describes the End of Mission (EoM) process for an L3 Area.

Description:

When a train completes a journey and the Driver closes the desk, the onboard issues an EoM request, and the train disconnects.

Performance indicators:

- *Completeness - probability that an object-status remains unknown/null (reliability & safety, quantitative);*
- *Reliability - probability that a vital object state value is wrong (reliability, quantitative).*

8.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

8.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

9) Operational Scenario #9 - Supervising Distance in Normal VCTS Driving

Abstract:

Supervision of train separation of a Virtual Coupled Train Set (VCTS) during normal driving.

Description:

This operational scenario addresses the supervision of train separation distance during normal driving in Virtual Coupling, and specifically, it assumes that Virtual Coupling has been already initiated.

The scenario starts with a VCTS (made of at least two trains) running under nominal Virtual Coupling conditions and aims at evaluating VCTS system safety and performance.

Performance indicators:

- *Probability of hazards due to positioning or communication faults – Probability of having an incorrect safe distance due to positioning errors or delay/errors/loss of communication (safety, quantitative)*
- *Line capacity – Measuring the expected increase of the line capacity compared to non-VCTS (performance, quantitative)*

9.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving					

Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

9.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

10) Operational Scenario #10 - Splitting of a VCTS Initiated by Slave

Abstract:

Termination of a Virtual Coupling session by splitting of a Virtual Coupled Train Set (VCTS) initiated by a slave.

Description:

This operational scenario addresses the termination of a Virtual Coupling session, and in particular the splitting of a VCTS initiated by a slave.

The scenario starts with a VCTS running under normal Virtual Coupling driving and ends with two VCTSs (possibly two standalone trains) running under Moving Block signalling.

Performance indicators:

- *Probability of collision – Probability that the relative distance between two trains in a virtual coupled train set becomes zero or less (safety, quantitative)*
- *Splitting time – The minimum time it takes for a slave train to split from a virtually coupled train set (performance, quantitative)*

10.1) Please, rate each of the following criteria for the described scenario.

	Very high	High	Medium	Low	Very low
Significance for the Moving Block signalling system					
Impact on the system safety					
Industrial relevance of the evaluation					

10.2) Do you have any suggestion on this scenario? (e.g., additional evaluations which increase the industrial relevance, possible variants/parameters)

11) Please, select your "best-4" (i.e., most meaningful) operational scenarios.

- ☐ OS#1 - Trackside Initialisation
- ☐ OS#2 - Start of Mission
- ☐ OS#3 - Points Control
- ☐ OS#4 - Crossing of a Radio Hole
- ☐ OS#5 - Loss/Restore of Communications

- ☐ OS#6 - Loss of Train Integrity
- ☐ OS#7 - Shunting Movement
- ☐ OS#8 - End of Mission
- ☐ OS#9 - Supervising Distance in Normal VCTS Driving
- ☐ OS#10 - Splitting of a VCTS Initiated by Slave