Deliverable 2.2

Moving Block Signalling System Test Methods

Project acronym:	MOVINGRAIL
Starting date:	1/12/18
Duration (in months):	25
Call (part) identifier:	H2020-S2R-OC-IP2-2018
Grant agreement no:	826347
Due date of deliverable:	30/06/2020
Actual submission date:	14/07/2020
Responsible/Author:	Gemma Nicholson (UoB)
Dissemination level:	PU
Status:	Issued

Reviewed: yes

Document history		
Revision	*Date*	*Description*
0.1	08/06/2020	First draft
0.2	01/07/2020	Final draft incorporating advisory board comments
1.0	03/07/2020	Final issue after quality check
2.0	11/09/2020	Incorporation of minor modifications requested by Commission

Report contributors		
Name	*Beneficiary Short Name*	*Details of contribution*
Achila Mazini	UoB	Test architecture and test automation strategy
Mohamed Samra	UoB	Background and test architecture
Gemma Nicholson	UoB	Overview, planning
Lei Chen	UoB	Review
Rob Goverde	TU Delft	Review and quality check

Funding

This project has received funding from the Shift2Rail Joint Undertaking (JU) under grant agreement No 826347. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Shift2Rail JU members other than the Union.

Contents

Executive Summary

This document is Deliverable 2.2 (D2.2) of MOVINGRAIL, “D2.2: Moving Block Signalling System Test Methods”. It is the second of two deliverables in WP2 – Approaches for Testing Moving Block Signalling, whose objective is to define improved strategies and methods for testing moving block systems, and to consider the trade-offs between laboratory and on-site testing. The first WP2 deliverable, D2.1 Moving Block System Test Strategy, identified the main stakeholder functional requirements for a moving block testing system, and developed a set of operational concepts.

The aim of this deliverable is to outline an architecture and testing routines for moving block signalling systems that builds upon D2.1 and other relevant sources. This necessitates the identification of the differences between ETCS-L2 and moving block systems at architecture level, and then of the gap between the established testing procedures for ETCS-L2 and the requirements that have been identified for testing of moving block signalling systems.

The distinguishing characteristic of ETCS-L3, which can be defined with virtual fixed blocks or as a full moving block system, is the fact that train detection no longer relies on trackside equipment. Instead, the safety-critical task of determining whether a section of track is occupied is performed by the on-board computer and the Radio Block Centre (RBC). The integrity of trains, i.e. that there are no loose carriages or wagons, must be verified by train-based rather than trackside systems. This is especially important as trains run closer together.

This document is divided into two parts. Part A presents the modular testing architecture for moving block systems, including the required components and interfaces. Part B uses the knowledge acquired from Part A and MOVINGRAIL D2.1 to present approaches for automated testing and to remove the current technical limitations for testing moving block systems. Strategy and automation approaches to minimise moving block on-site testing are identified.

The work of X2RAIL-1 on the specification of moving block signalling systems at sub-system and architectural levels feeds into the identification of signalling system architecture differences (ETCS-L2 vs moving block). This allows the development of a testing architecture and associated interfaces for a moving block system that builds upon the testing systems for ETCS-L2 signalling.

The testing system architecture specifies the components and interfaces of an integrated system that can test system behaviour for moving block systems regardless of system supplier. The testing architecture is backwards compatible, i.e. it applies to levels L1 and L2 of ETCS, and is extensible to the testing of future virtual coupling systems. The architecture can be divided into four groups: i) a test control and logging unit to control and monitor the tests, ii) on-board constituent components, iii) simulators or hardware-in-the-loop components for external including trackside entities, and iv) additional components. The main components are specified in more detail: functionality and interface to the TCL and other relevant component(s).

Part B describes the requirements for a test automation strategy that minimises field testing. The high-level strategy consists of the following components: scope identification, testing framework and design approach, test script creation, test execution, and automated analysis. Metrics for testing the effectiveness of test automation are discussed and a movement authority allocation scenario is discussed in context.

Abbreviations and Acronyms

Abbreviation/Acronym	Description
ARS	Automatic Route Setting
ATP	Automatic Train Protection
BMC	Bounded Model Checking
BTM	Balise Transmission Module
CAV	Controlled Autonomous Vehicle
CIT	Combinatorial Interaction Test case
CMD	Cold Movement Detection
DMI	Drive Machine Interface
DNN	Deep Neural Network
ERTMS	European Rail Traffic Management System
ETCS	European Train Control System
EVC	European Vital Computer
EoA	End of Authority
FFFIS	Form Fit Function Interface Specification
GCG	Ground Communication Gateway
HIL	Hardware-in-the-loop
HMI	Human Machine Interface
IXS	Interlocking
JRU	Juridical Recording Unit
KMC	Key Management Centre
LTM	Loop Transmission Module
MA	Movement Authority
MB	Moving Block
MCG	Mobile Communication Gateway
OBU	On-Board Unit
ODO	Odometry Unit
RBC	Radio Block Centre
RBS	Radio Base System
RBT	Requirements Based Testing
RIU	Radio Infill Unit
RTM	Radio Transmission Module
SRS	Software Requirements Specifications
STM	Specific Transmission Module
SUT	System Under Test
TCL	Test Control and Logging
TCMS	Train Control & Monitoring system
TCS	Train Control System
TIU	Train Interface Unit
TMS	Traffic Management System
VBTS	Virtual Balise Transmission System
ZOST	Zero On-site Testing

1. Introduction

This document is Deliverable 2.2 (D2.2) of MOVINGRAIL and it is part of WP2: Approaches for Testing Moving Block Signalling. The aim of MOVINGRAIL is the further development of train-centric signalling systems by introducing and applying a multidimensional analysis framework to assess train-centric signalling from the operational, technological and business perspectives. WP2 addresses the testing approaches necessary for the implementation of moving block signalling. The objective is to define improved strategies and methods for testing moving block signalling systems, including trade-offs between laboratory and on-site testing.

This document is divided into two parts. Part A presents a modular architecture and the required components and interfaces for moving block testing; Part B uses the knowledge acquired from the previous part and Deliverable D2.1 to present approaches that support automated testing and remove the current technical limitations for testing moving block systems.

1.1. Scope

Train control system (TCS) is a broad concept that can be described as a group of railway hardware devices and subsystems that monitor and manage a train's operation based on its location, headway, speed and route, not only to prevent collisions and derailment but also to improve network operations [1]. Many types of train control systems have been developed and introduced to date. These systems vary depending on configuration and the type of safety solution used and can range from a simple automatic warning system to the most advanced communication-based train control system.

The European Rail Traffic Management System, also referred to as ERTMS, was specified by the International Union of Railways (UIC) and several manufacturers and offers a uniform signalling system for seamless cross-border operations for high speed and conventional trains as well as freight hauling [2]. At level 3, ETCS may be a moving block system: it offers solutions with full moving block and fixed virtual block system types, with or without trackside train detection equipment. The relevant requirements and definitions regarding the system that have been adopted in this document were previously defined in X2Rail-1 [3] and X2Rail-3 [4] projects, which are cited in the text. The following documents were consulted in the development of this deliverable:

- X2RAIL-2, Deliverable D3.1, System Requirement Specification of the Fail-Safe Train Positioning Functional Block, 2019 [5].
- X2Rail-2, Deliverable D4.1, Train Integrity Concept and Functional Requirement Specification, 2017 [6].
- X2Rail-2, Deliverable D6.3, Description of Use-Cases for new TMS Principles, 2018 [7].
- X2Rail-1, Deliverable D3.3, Specification of the communication system and Guideline for choice of Technology, 2017 [8].
- X2Rail-1, Deliverable D5.1, 'Moving Block System Specification', 2019 [9].
- X2Rail-1, Deliverable D6.1 Current test condition and Benchmarking report, 2017 [10].

This document also requires MOVINGRAIL D2.1 as input, since the analysis carried out assumes the validity of the specifications and requirements defined in that document.

This deliverable presents an overall modular architecture and routings for testing moving block signalling systems that is also expandable to allow the virtual coupling concept to be tested. The scope in this deliverable is on the functional and data tests for moving block systems from suppliers developed for specific engineering projects. In Part A, the report identifies an extensible test architecture highlighting the key components and interfaces defined between them. Besides, the section discusses the component functionalities under a finite number of states to allow components to be type-tested using conventional methods. Moreover, Part B presents a comprehensive systematic approach for testing the correctness of the configuration data when assigning a movement authority under the moving block system. This is followed by suggesting strategy and automation approaches that aim to minimise on-site testing for moving block systems.

2. Background

The European Rail Traffic Management System (ERTMS) can be divided into two major subsystems: (i) global system for mobile communications provided via a radio system also known as GSM-R and (ii) signalling and train protection systems, also known as the European Train Control System (ETCS), which covers the on-board and trackside components.

Conventionally, train detection and movement authority are provided via trackside equipment. In this regime, train detection sections are strategically positioned to divide the line into limited sections, also called fixed blocks. Either lineside signals or movement authorities received on-board authorise trains to enter these sections.

Nowadays, most worldwide implementations of ERTMS are of ETCS level 1 and level 2 systems [11], which still rely on trackside train detection technology. Of these, ETCS level 2 is the one which can provide greatest improvements in capacity. On the other hand, ETCS level 3 is emerging as a solution to allow the current system to take a step further towards improved performance. Considering its potential to eliminate the dependency on trackside train detection, ETCS-L3 can improve the capacity and flexibility of railways and reduce infrastructure and maintenance costs.

In contrast to predecessor systems, ETCS-L3 blocks are not defined by pairs of adjacent fixed points on the line, usually at signals. Rather, their establishment is made by either (i) virtual fixed block system – using a type of block that mimics the operation of a fixed block system, where a movement authority is issued to a fixed point on the track or (ii) moving block system – where the blocks are defined by a computer system and move with trains as they travel along the track. As a result, trains can be given authority to move to any location on the track, allowing train separation based on the absolute braking distance between trains.

Figure 1 presents an overview of ETCS-L3 highlighting the differences from ETCS-L2 in terms of on-board (left) and trackside components (right). In ETCS-L3, the potential absence of trackside train detection enables very short fixed virtual block sections by adapting the configuration of the trackside system, or even full moving block sections, thus improving the whole system performance at a relatively low cost [12].

Different approaches have been proposed to provide train positioning without the dependency on trackside equipment such as a Virtual Balise Transmission System (VBTS), enhancing the odometry accuracy, or a combination of both. Migration from trackside balises is expected to reduce the CAPEX and OPEX as well as reducing the time intervals between consecutive resets of train position confidence interval as the virtual balises are expected to be more frequent on the track compared with the current physical balises [8]. These new approaches might define the scope of the future train positioning system in ETCS-L3. Currently, the train positioning of ETCS-L3 will continue using the Balise Transmission Module as in ETCS-L2 until a future system is applied.

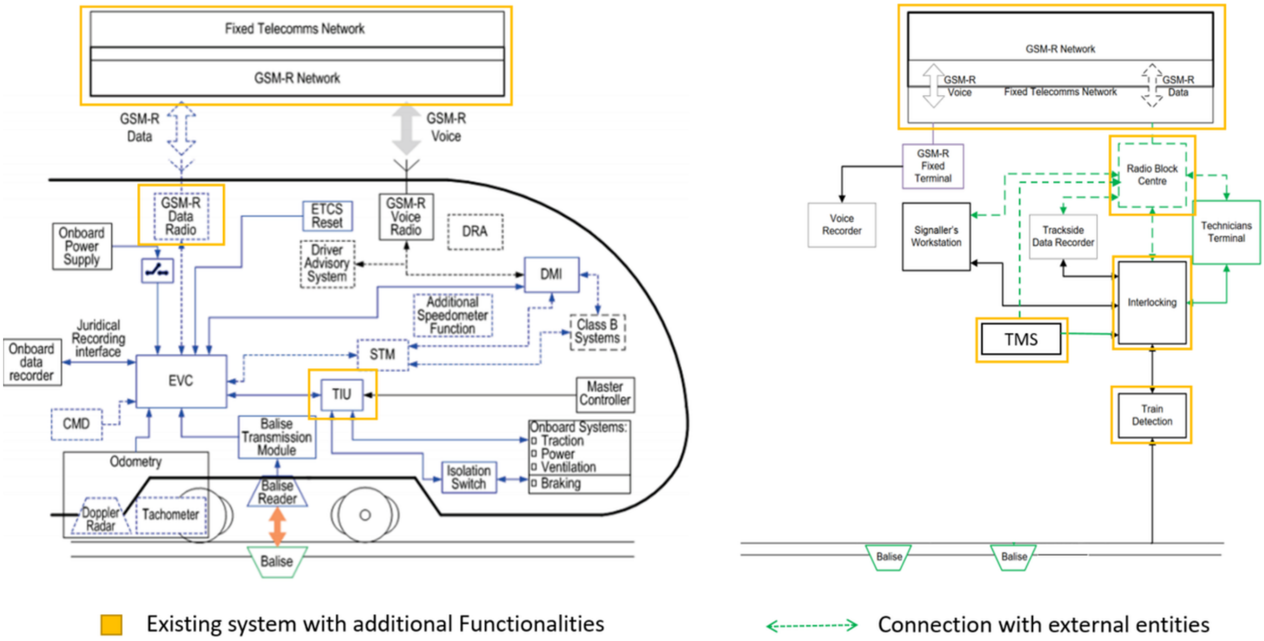


Figure 1: ETCS-L3 system overview

Furthermore, ETCS-L3 deploys the train integrity function differently by deploying an additional subsystem to the Train Interface Unit (TIU) which is the "Train Integrity Proving System" rather than depending on trackside equipment to provide the train integrity functionality as in ETCS-L2 [6]. In ETCS-L3, the on-board train integrity monitoring unit is responsible for verifying the completeness of the train while the train is in operation. The functionality focuses on ensuring the coherent movement of the last wagon in relation to the remainder of the train. An accidental wagon decoupling can result in serious accidents by forming unexpected obstacles on the track [6]. The train integrity functionality is part of the Train Interface Unit (TIU), demonstrated in Figure 1.

X2RAIL-1 D3.3 [8], states that the on-board communication systems of ETCS-L2 and ETCS-L3 will still have the same functional requirements as one another, as defined in [14]. However, some additional functions for the trackside system are expected to arise due to new scenarios introduced in ETCS-L3 such as loss of communication, movement of a non-communicating train, recovery management after loss of communication, and management of a radio hole [9]. Section 3.2.2.4 of this report highlights these additional functionalities. Furthermore, with the current communication system limitations, there is industrial direction to adopt an altered communication system architecture to support transparency of networks in use, concurrent use of communication bearers, vertical handover management between bearers, support of end-to-end application security, and capability to use and select a bearer based on the application requirements. The prospect of altered architecture of the on-board and trackside communication components will not change the main functionalities that are currently provided in ETCS-L2.

Trackside train detection is optional for ETCS-L3 operations. However, the trackside train detection system in ETCS-L3 aims to faster release points and crossings as well as allow faster recovery from degraded situations [9]. In addition, there are some situations that might require trackside detection for certain locations or areas of control. For instance, the train detection system can be used to detect and permit movement of trains which are not equipped with train integrity systems

in a mixed traffic scheme. Also, the trackside train detection system can detect the entry of unfitted trains, unexpected or unauthorised movement at critical points, e.g. switches, and entry of other non-communicating vehicles to the area of control (e.g. shunt operations).

The system architecture in the baseline 3 release 2 ETCS specification [16] does not consider the interlocking as part of the ETCS system. This document focuses on the ETCS-L3 trackside component without focusing on the actual interface between the interlocking and RBC.

Traffic Management System (TMS) is responsible for monitoring, controlling, and commanding the traffic and the signalling systems from the control centres. The TMS is considered to be an external system to the trackside system, with several actors involved in various scenarios [7]. In ETCS-L3, the TMS must be able to correspond with the various ETCS-L3 systems to include full moving block system types, with and without trackside train detection as well as fixed virtual blocks with and without trackside train detection. Examples of ETCS-L3 TMS functionalities can be found in [7], while a prototype of the interface between the TMS and ETCS-L3 trackside is defined in [17].

Regardless of the different solution types for modern train control systems, all of them have the ultimate goal of improving safety, monitoring train integrity, increasing efficiency and capacity, and providing a standardized and interoperable railway. Testing plays an important role to check the safety and reliability of the system before the system is deployed.

However, applying testing techniques to train control systems can be considered a difficult and time-consuming task due to the system characteristics of high complexity and non-determinism [18]. Despite many approaches available in the railway industry, up to now, ETCS testing and validation is a long process that must take place before systems can be put into service [19].

Train control systems are highly integrated systems with a large number of different functions, which are reflected in the complexity in generating efficient models and simulations to represent the system. As a result, the approach to test a TCS needs to consider a combination of simulation technology and testing technology. In this context, the simulation creates a representative environment that emulates in the most accurate way real-world operation, while the testing technology takes into account the testing process and state-of-the art automation techniques.

Moreover, in order to describe a complex system and more specifically the railway system, two elements are important: the system architecture and the prediction of its behaviour. The system architecture should define a hierarchical decomposition of the components of the System Under Test (SUT) and the system's functions and requirements. The prediction of its behaviour is shown as a model that is able to illustrate and represent a full range of reasonably-expected operational scenarios.

The challenge is then to test that the system fulfils the safety requirements. For moving block there is the modified situation regarding train integrity monitoring e.g. the unintentional decoupling of wagons or accidentally leaving vehicles on a main line after shunting operations have been carried out. The moving block and virtual coupling paradigms are associated with high operational complexity since they create an infinite number of movement authority allocation scenarios in contrast to the fixed block case. An effective way to approach this challenge is by applying test automation strategies and methods designed for testing moving block signalling systems, focusing on a trade-off between laboratory and on-site testing.

As determined from stakeholder interviews and the literature review presented in *D2.1 Moving Block Signalling System Testing Strategy* [20] , the key point to achieve a robust testing solution for moving block systems relies on increasing the reliability of current laboratory test practices and providing a test environment that is able to reproduce relevant operational scenarios and conditions as realistically as possible. These premises form the basic assumptions that will be investigated in this document.

PART A: ARCHITECTURE

3. Extensible architecture

This section provides an overall modular architecture suitable for application to the functional and data testing of future moving block signalling systems for specific projects. It is also appropriate for testing the current ETCS-L2 and ETCS-L1. This document has been developed by making reference to other publications outlining the appropriate established standards and therefore incorporates some provisions from them, which are cited at the appropriate places in the text.

Deliverable D2.1 of the MOVINGRAIL project titled "Moving Block Signalling System Test Strategy" [20] identified the key stakeholder functional requirements, reviewed testing methods of safety-critical systems, before developing operational concepts and a system testing strategy for moving block systems. The output of that deliverable has been incorporated into this document to define best practice in safety critical system testing and hardware-in-the-loop testing.

UNISIG Subset 026 [21] specifies ETCS from a technical perspective by defining the rules for messages to and from trains, i.e. between on-board and trackside systems; they have been captured in this document.

The ETCS reference test facility to perform tests on the on-board components is defined in UNISIG Subset-94 [22]. UNISIG Subset-111 [23] specifies a general test environment to test ETCS-L2. It presents a test architecture that enables the test sequences in Subset-076-6-3 to be executed [24].

Subset 037, EuroRadio FIS [14] defines the message exchange between the on-board and trackside equipment for ETCS radio system interoperability. The output of this subset has been used throughout this document to highlight the main functional requirements of the EuroRadio. Finally, UNSIG SUBSET-041 [25] analyses the required technical performance of the ETCS equipment that are relevant for interoperability. The interoperable interfaces include the DMI, the train interface, the EuroRadio interfaces, and the Eurobalise air gap.

3.1 Design principles

The design principles derived from the requirements and recommendations presented in D2.1 Moving Block Signalling System Test Strategy and SUBSET-111 [23] are as follows:

a. The architecture has to be adaptable to support ETCS L1, L2, and L3 configurations.

b. The architecture has to be capable of upgrade to support virtual coupling and future versions of ERTMS/ETCS.

c. The architecture must be simple and allow test assemblies and components from different projects or suppliers.

d. The specification of components and interfaces shall cover only commonly required functionalities.

e. The use of hardware-in-the-loop (HIL) solutions shall be implemented to achieve close to real operation as much as possible.

f. The implementation of the functions in the adaptor is the responsibility of each company.

3.2. Architecture overview

The system architecture shown in Figure 2: has been designed to provide a single integrated system capable of accommodating and testing different suppliers' trackside and on-board ETCS products based on ERTMS/ETCS operational rules and procedures. The architecture is considered an update of the test architecture of subset 111 [23] as it includes specific components needed to enable a moving block system including the train integrity as part of the TIU as well as the VBTS for train positioning.

This architecture can also accommodate the testing of all levels of ETCS, including future moving block signalling system specifications and, allows further expansion to test virtual coupling. Furthermore, this architecture allows the simulation of the necessary interacting parts of a complete railway system with simulated train movements, by having testing inputs, outputs and the behaviour of ETCS sub-systems fully simulated as described in Deliverable 2.1 *Moving Block Signalling System Test Strategy* [20].

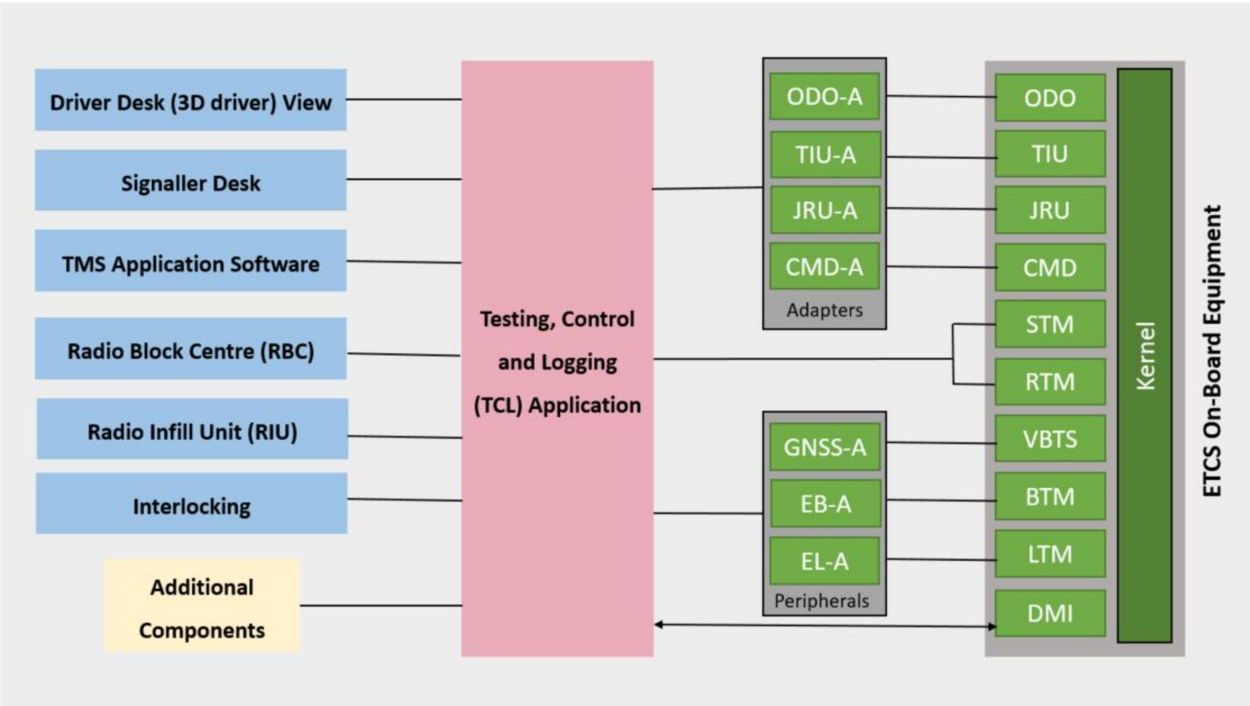


Figure 2: ERTMS Test Architecture

In Figure 2:, the testing environment is divided into four main groups:

(i) Pink box: which represents the test execution domain containing the Test Control and Logging (TCL) unit;

(ii) Green boxes: indicate the on-board constituent components. The represented ERTMS/ETCS constituent has the potential to be extended to include more modules;

(iii) Blue boxes: representing the external entities to be integrated in the test environment such as simulators or any hardware in the loop (HIL) components;

(iv) Yellow box: indicating any additional components or external tools that can be attached to the architecture as per the stakeholder's needs.

This architecture must include (unless described otherwise as optional), but is not limited to, the components discussed in the following sections, which have been collected into testing in Section 3.2.1, external entities in Section 3.2.2, and on-board domains in Section 3.2.3.

3.2.1. Testing domain

3.2.1.1. Testing, Control and Logging (TCL) application

The Testing, Control, and Logging Application (TCL) is the central component in a testing environment that manages the simulation and testing configurations and processes to represent the state of the railway network and traffic. It acts as a central unit between the external simulators, HIL components, and the ERTMS/ETCS on-board constituent.

In addition, the TCL shall interact with the ERTMS/ETCS on-board equipment through the interfaces defined in Subset-026, Section 2.5.3 titled “ERTMS/ETCS Reference Architecture” [26]. This document summarises the TCL interfaces and functionalities. Further details can be found in SUBSET-111 [23].

The TCL entity is capable of initiating several distinct functions such as providing multi-train simulation, occupying and clearing train detections based on train movement, feeding coherent train position, speed, and acceleration to the odometry via the intermediate interface, logging data, checking train integrity, generating DMI actions, and simulating a complete railway system with mixed traffic operation (i.e. freight and passenger) and mixed signalling types (i.e. ETCS-fitted and unfitted trains).

Besides, the TCL application should allow the setup, execution and analysis of the test scenarios specified in Subset-076-6-3 [24]. The TCL should be able to fulfil the testing objectives of different stakeholders and provide a compliant Form Fit Function Interface Specification (FFFIS), where defined. The next sections of this document highlight the distinct functionalities of the TCL entity when connected to the components showed in Figure 2.

3.2.2. External entities domain

3.2.2.1. Traffic Management System (TMS)

The traffic Management System (TMS) application is connected to the TCL to manage timetables and enable multiple train simulation. The TMS should be able to implement basic features that include:

(i) Train tracking
(ii) Interlocking commands and/or Automatic Route Setting (ARS)
(iii) Infrastructure controls (e.g. traction system, reduced adhesion areas)
(iv) Train status
(v) Conflict detection and resolution

Under moving block operation, the TMS incorporates more functionality than the other levels of operation. These functionalities include [23]:

(i) Setting/clearing areas where the track status is UNKOWN
(ii) Moving points within a known area by emergency procedure
(iii) Providing an estimated train location to the trackside component

(iv) Providing the Staff Responsible (SR) distance to the trackside
(v) Validating the SR distance calculated by the trackside
(vi) Activating and disabling shunting areas
(vii) Establishing and removing temporary shunting areas
(viii) Activating and deactivating dynamic radio holes

The high-level functional requirements of the TMS can be found in [27], while moving block focused requirements are detailed in [9].

3.2.2.1.1. TCL – TMS Interface

The architecture to interface the TMS with the TCL is shown in Figure 3:. The TCL should be able to interface with any TMS provided by an appropriate supplier. In Figure 3, the TCL forwards the static variables including the network topology and timetable information as well as the dynamic variables originated from the RBC/interlocking to the TMS. These dynamic variables include the position updates for all trains in the control area being monitored by the TMS. Additionally, TMS sends back the control plans to the TCL including the signalling instructions. Another functionality of the TCL is logging the simulation scenario and the passed variables.

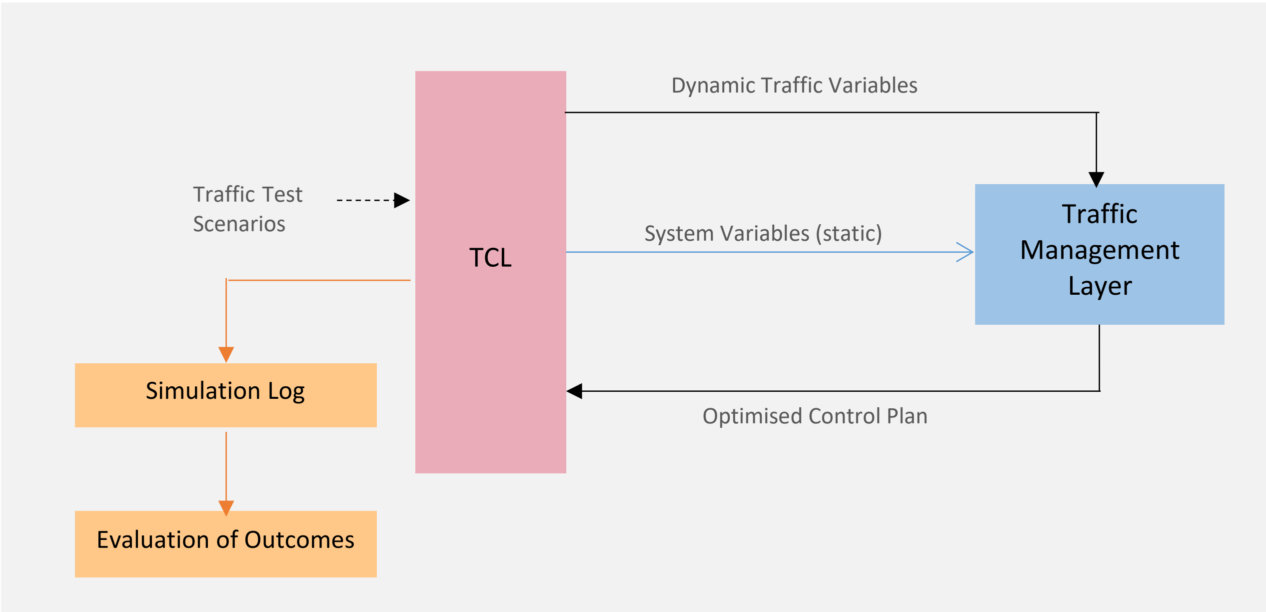


Figure 3: TMS framework

3.2.2.2. Signaller Desk

The signaller desk is a system that simulates the train traffic control by signallers and dispatchers with a similar UI to a real traffic control room. The signaller desk shall include, but not be limited to, the following functionalities:

(i) Traffic display
(ii) Interface to set routes, including sanity check and configurable interlocking timings
(iii) Interface to unset routes, including configurable approach locking

3.2.2.2.1. TCL – Signaller Desk Interface

The signaller desk interface, represented in Figure 4, requires a large amount of information that is to flow between the TCL application and the signaller desk which includes:

(i) Traffic state including the speed and location of all of the trains in the signaller's area
(ii) Interlocking state, including route setting, point settings, and signal aspects (based on the ETCS level being tested)
(iii) Train movement authorities.

The TCL works on forwarding the needed information from RBC/Interlocking and TMS to the signaller desk as well as logging the simulation data.

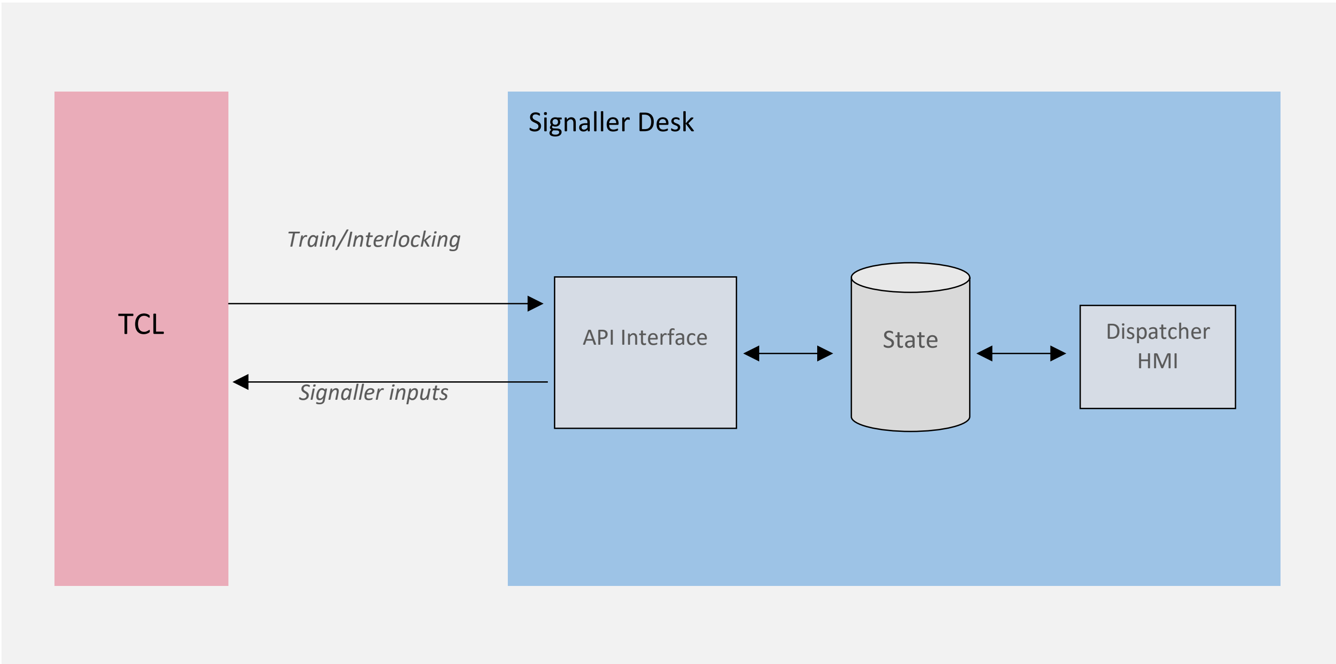


Figure 4: Signaller desk interface

3.2.2.3. Driver Desk with 3D Views

The driver desk is an optional element that is able to provide a 3D graphics view of the simulated railway environment. The driver desk consists of two separate components with separate interfaces:

(i) The Driver 3D View which represents the driver's view outside of the train's window
(ii) The Cab controller that consists of a number of sub-components, with data flowing both to and from the TCL component:
   - A sound controller
   - A hardware controller

3.2.2.3.1. TCL – Driver Desk/3D View Interface

A driver desk consists of two components with separate User Datagram Protocols (UDP) interfaces, as shown in the example desk in Figure 5. Each component has a local state object that contains the contextualised state of the train which the driver desk represents. The TCL works on forwarding the RBC/Interlocking data to the driver desk and it logs all of the simulation data.

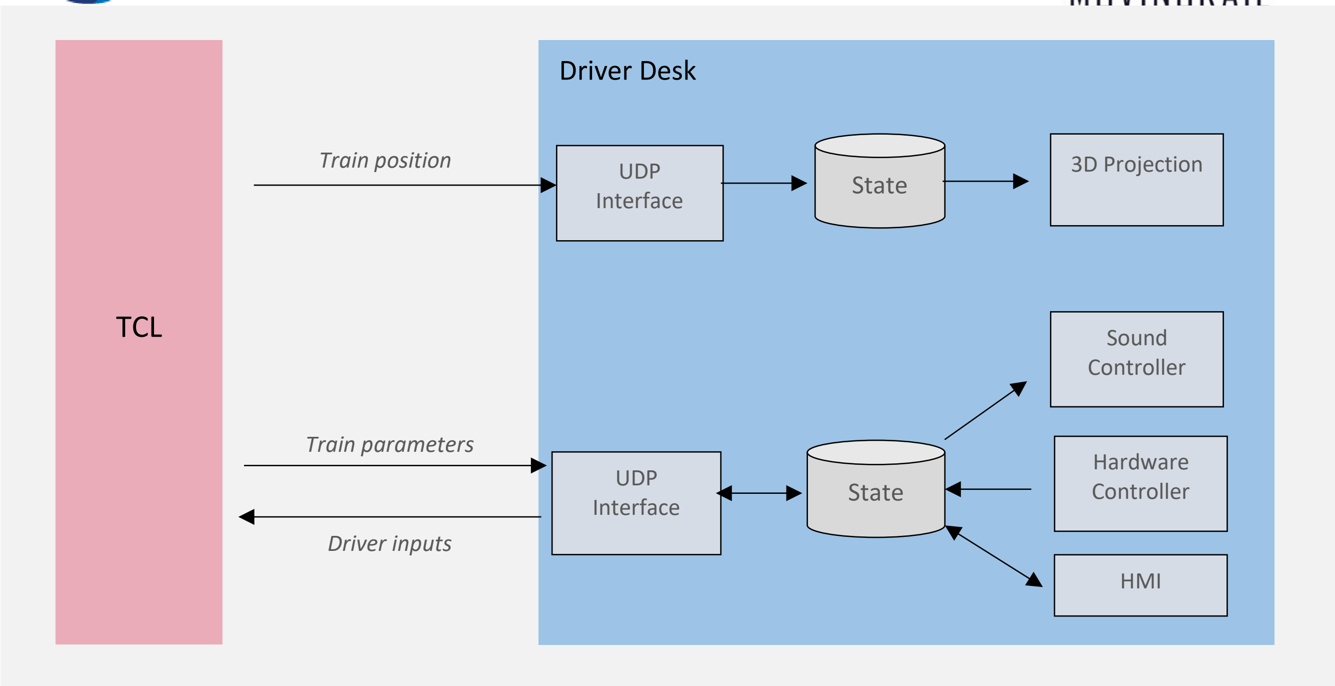


Figure 5: Driver desk architecture

3.2.2.4. Radio Block Centre (RBC)

The Radio Block Centre (RBC) is a computer-based system that manages the exchange of data between on-board and trackside. The exchanged data is used to provide movement authorities that allow the safe movement of trains on the railway infrastructure area under the responsibility of the RBC [21].

In an ETCS-L3 system, the movement authority is issued at most to the next obstruction which can either be at a fixed or a dynamic location. In a full moving block system, the movement authority can be issued to an arbitrary location that is determined by the rear of a preceding train. However, in a fixed virtual block system, the end of the movement authority is related to pre-determined locations at the boundaries of the virtual blocks.

The RBC in a moving block system imposes a new set of functionalities that include recovery management after loss of communication, movement of a non-communicating train, and management of a radio hole. Following loss of communication, the status of the track in front of the train is considered unknown as the train can be anywhere between the last confirmed safe rear end of the train and the most recent End of Authority (EoA). During this period, the driver can override to proceed under the staff responsible mode. If the communication session is restored, the track status can be recovered. If the communication sessions cannot be restored, then the track status will remain unknown. It is the duty of the RBC to define a mute timer after which the communication with the train is considered lost. Besides, the RBC should send a notification to the interlocking to set the track status as unknown [9].

The RBC in a moving block system should handle the movement of a train with failed communications. The RBC should enable a traffic management system to request a protected path for the affected train in areas of track status unknown. Finally, the RBC should be able to manage a radio hole that is either static (predefined since the RBC was commissioned (e.g. a tunnel)) or dynamic (occurs after the RBC was commissioned (e.g. base station failure)). The RBC in a moving block system executes two main functionalities to manage a radio hole [20]:

- Ensure that no MA is issued with an EoA within the radio hole.
- Store information regarding trains entering the radio hole, alert the traffic management system, and report that the train has not cleared a certain area after a pre-set timer expires.

The interface between radio communication systems providing communication services for safety-related applications can be found in SUBSET-037 [14], while [9] details the additional RBC functionalities introduced for a moving block system, and finally [28] presents the possibility of having one RBC under test and other simulated RBCs in the testing environment.

3.2.2.5. Radio Infill Unit (RIU)

As the introduced test architecture is expected to be extensible covering the various ETCS levels. This section covers the Radio Infill Unit (RIU) which is an optional component that is applicable to ETCS level 1 only. The RIU provides the signalling information in advance to the train by GSM-R transmission. In the test environment, the RIU corresponds to the real equipment provided by the supplier and no functional modifications compared to the commercial version are allowed.

3.2.2.5.1. TCL – RBC/RIU Interface

The RBC and RIU equipment will interface with the TCL via the API mechanism, as seen in Figure 6. The interface functionalities are detailed in the document SUBSET-111-3 [29] for TCL-RBC interface. Besides, the RBC/RIU Adaptor Unit enables the attachment of one or more RBC/RIU constituents to the test environment. More details can be found on SUBSET-111 [23].

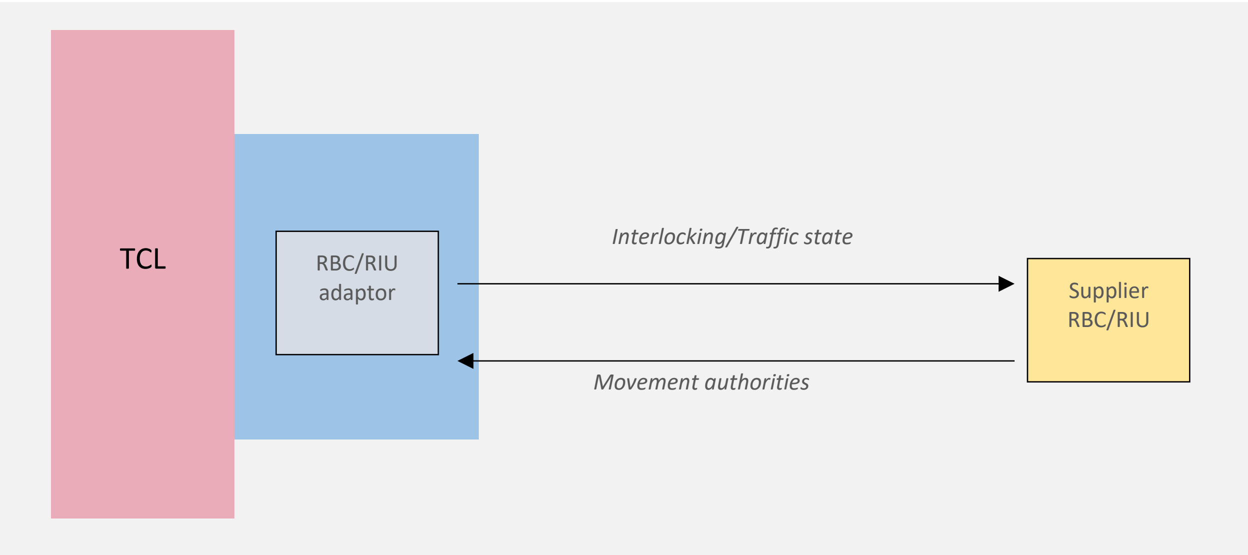


Figure 6: TCL-RBC interface

The RBC and RIU can be integrated with the on-board Radio Transmission Module (RTM) to constitute the Radio Base system (RBS). The RBS framework is part of the TCL layer and it is responsible for establishing a closed-down connection between its sub-systems while monitoring the message transfer between them, as shown in Figure 7. This framework is expandable to accommodate multiple RTM, RBC, and RIU units in the test environment.

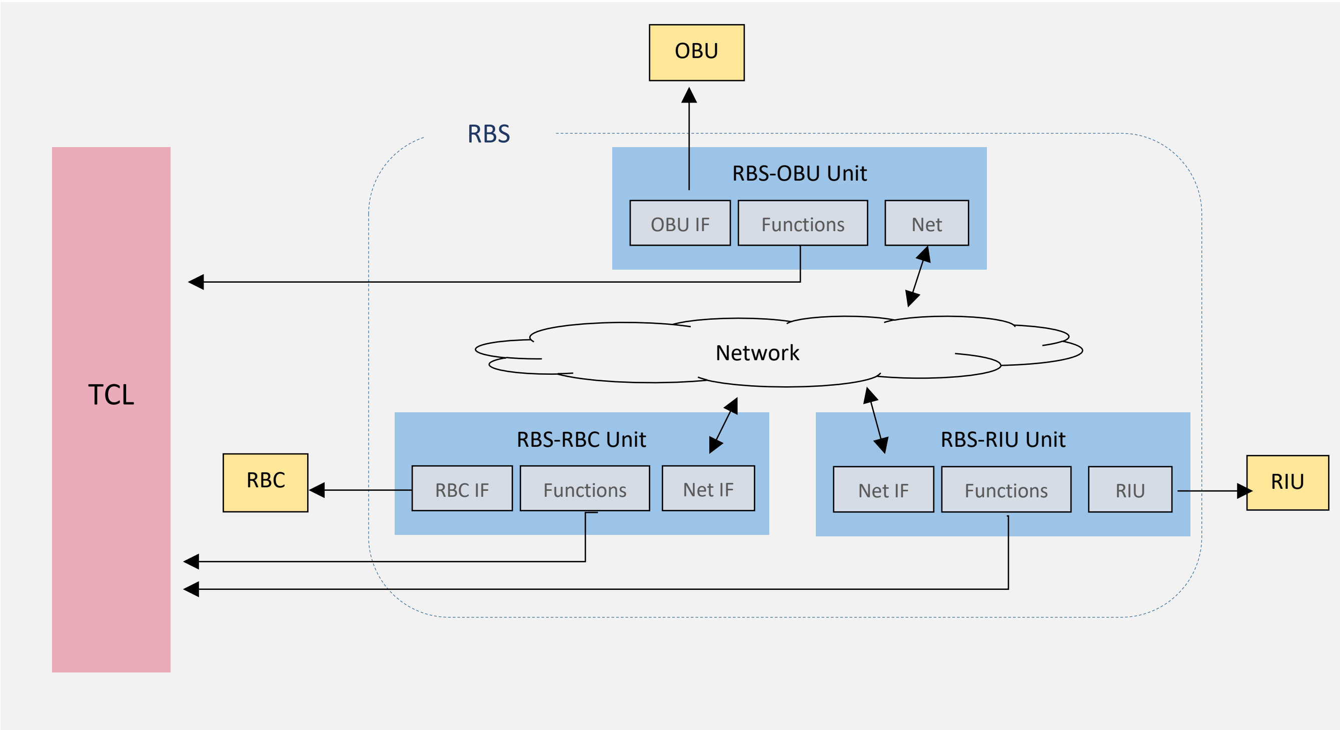


Figure 7: RBS interface

Composed for a combination of hardware and software elements, the main objective of the on-board unit (OBU) is to provide Automatic Train Protection (ATP). In conjunction with the ETCS trackside system and data entered by the driver, the OBU is able to monitor the position of the train and determine the safe envelope in which the train can move. Should the train exceed this envelope, then the on-board unit is able to intervene to control the speed of the train and bring it to a stand [30].

In the test environment, the OBU consists of modules that can interface with the TCL defined above in Figure 2. The following section will focus on defining the TCL-OBU interfaces highlighting the TCL functionality in case of OBU absence.

3.2.3.1. Odometry Unit (ODO)

The on-board odometry is linked to the TCL application through an adapter named ODO-A. The TCL layer and ODO-A can either generate multiple signals to satisfy the ODO in the EVC that the train is moving with the desired speed, acceleration and location uncertainty. Alternatively, the ODO-A may feed the desired speed, acceleration, and location uncertainty to the EVC core directly, replacing the ODO hardware. Another approach is to type-test the ODO unit itself without considering the TCL and ODO-A. The communication between the TCL and the on-board ODO is unidirectional, as shown in Figure 8. The physical characteristics of this interface can be found in [22].

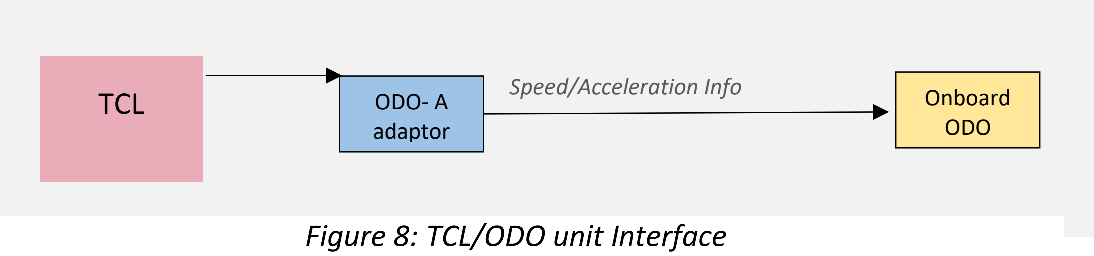


Figure 8: TCL/ODO unit Interface

3.2.3.2. Train Interface Unit (TIU)

The TCL has a bi-directional communication with the on-board TIU through the TIU adapter (i.e. TIU-A), as shown in Figure 9. The TCL should be able to forward and log data passed to the TIU from the EVC. The physical characteristics of TIU-A are defined in [22]. The communication between the TCL and TIU should adhere to all the train interface information presented in [26] and [31]. For instance, the TCL can have a different set of inputs for the TIU that are categorised into dependent and independent inputs. Special brake status and traction status are examples of the dependent inputs, while train integrity and train data are examples of independent inputs retrieved from external units.

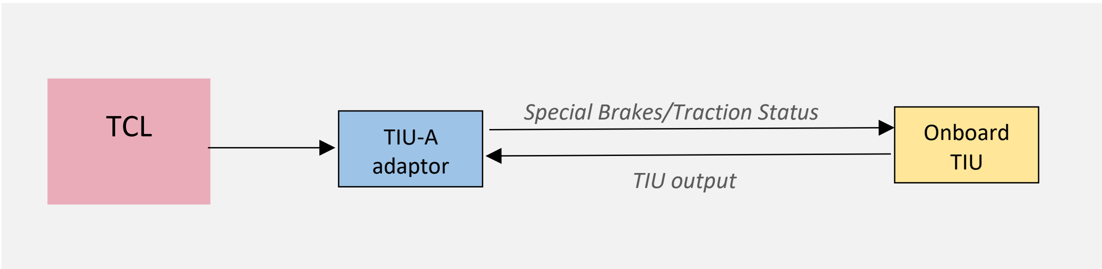


Figure 9: TCL/TIU unit Interface

Monitoring the status of the train's tail to ensure movement coherence is a key feature in the On-board Train Integrity (OTI) needed for the moving block concept. The OTI is an additional subsystem added to the on-board TIU to eliminate the need for trackside detection equipment to perform the train integrity functionality as in ETCS-L2. The OTI is responsible for handling the communication between the master and slave modules to ensure that the train tail status is determined and monitored. X2RAIL-2 D4.1 titled "Train Integrity and Functional Requirements Specifications" details the interface functional requirements between the OTI and on-board TIU [6].

3.2.3.3. Juridical Recording Unit (JRU)

The communication between the JRU and TCL is unidirectional through the intermediate adapter (i.e. JRU-A). The communication diagram is identical to the ODO-A shown in Figure 7. The main role of the TCL is to forward and log data passed to the JRU from the EVC.

3.2.3.4. Cold Movement Detection (CMD)

The communication between the CMD and TCL is unidirectional through the intermediate adapter (i.e. CMD-A). The communication diagram is identical to the ODO-A shown in Figure 7. The TCL should provide information to the on-board CMD unit such as: not available, train has moved, train has not moved, or fail state. The physical characteristics of this adapter can be found in [22].

3.2.3.5. Specific Transmission Module (STM)

The TCL shall manage a bidirectional communication with the on-board STM equipment. The messages should fulfil what is described in [32]. The TCL is responsible for applying the protocols of [33] and [34] to encode/decode message exchanges with the on-board equipment.

3.2.3.6. Radio Transmission Module (RTM)

The RBS is a modular component of the TCL Layer that interfaces with the on-board RTM to execute two main functionalities. The RBS module should be able to imitate a trackside safe application such as an RBC or a radio infill unit in case there is no external unit is connected. The second functionality of the RBS is to emulate at least two Class B GSM-R mobile terminals at physical and logical levels and it needs to be compliant with the procedures detailed in [14] and [35]. More details on the TCL functional interface with the RTM can be found in [22].

3.2.3.7. Virtual Balise Transmission System (VBTS)

The VBTS is considered to be a promising approach that might replace the current Balise Transmission Module for train positioning. As the system is seen as only a prospect system, it was not added to Figure 2. This section gives a brief introduction about the system and the main functionalities it is expected to perform. Satellite positioning was considered one of the key elements in future railway signalling systems (i.e. moving block systems) that contributes to the reduction of CAPEX and OPEX associated with the physical Balises. The X2RAIL-2 project has looked into the introduction of an additional on-board transmission system named Virtual Balise Transmission System (VBTS) [5]. The main functionalities of the VBTS include handling GNSS signals, determining pseudo ranges, estimating safe train positioning, integrating data from other kinematic sensors, and detection of a virtual balise. The TCL layer should interface with the VBTS and it should be able to generate a virtual balise telegram that is compliant with X2RAIL-2 D3.1, ‘System Requirement Specification of the Fail-Safe Train Positioning Functional Block’ [5]. Also, the TCL should able to manage a list of virtual balise telegrams that is either loaded in the configuration phase or updated dynamically as the simulation runs.

3.2.3.8. Balise Transmission Module (BTM)

the TCL interfaces with the on-board BTM and it should be able to generate a balise telegram that is compliant with [36]. Also, the TCL should able to manage a list of balise telegrams that is either loaded in the configuration phase or updated dynamically as the simulation runs. The list of these telegrams should contain the balise location. Either this list is arranged by a single balise or balise groups. More details on the TCL functional interface with the BTM can be found in [22].

3.2.3.9. Loop Transmission Module (LTM)

For an ETCS-L1 system, the TCL interfaces with the on-board LTM and it should be able to generate a Euroloop message that is compliant with [37]. Also, the TCL should able to manage a list of Euroloop messages that is either loaded in the configuration phase or updated dynamically as the simulation runs. The list of these messages should contain the Euroloop start and end location to be used for the correct loop signal generation. More details on the TCL functional interface with the LTM can be found in [22].

3.2.3.10. Driver Machine Interface (DMI)

The TCL should interface with the DMI to record every input and output on the DMI device in a digital format. Optionally, the TCL can automate the DMI inputs without manual staff interventions

to execute a certain test sequence. More details on the TCL functional interface with the DMI can be found in [22].

3.3 Summary of moving block system test architecture and interfaces

This section presented a modular architecture which is suitable for the testing of future moving block signalling systems. The proposed architecture highlights the extended functionalities of existing sub-systems, including the TMS and the RBC, needed for moving block system operations. Furthermore, the architecture listed the newly added sub-systems that are unique for ETCS-L3 such as the train integrity (as part of the TIU) and prospect systems such as the VBTS for more accurate train positioning. It is expected that the architecture will be also appropriate for the current ETCS-L2 and ETCS-L1. To achieve this, the architecture has maintained the components that are needed for ETCS-L1 such as the LTM and RIU. A scenario is discussed in the next section of this document to demonstrate how the extensible architecture is applicable to a general scenario of the moving block system.

PART B: TESTING METHODS

4. Introduction

This section describes a set of high-level approaches to support automated testing in the laboratory in order to minimise on-site testing for moving block systems.

Nowadays, the majority of companies perform both laboratory tests and on-site tests. In fact, on-site tests are still required since no industry has shown the capacity and capability to perform all relevant tests in laboratory [10].

This reality is not different in the railway sector. Even though many tests have been shifted to laboratories through the years, a significant number of tests have historically been performed on-site. As a matter of fact, there are considerable technical limitations to address when discussing the approaches towards zero on-site testing. There are operational and system characteristics intrinsic to ETCS that are impractical to reproduce digitally or simulate and cannot be performed in a laboratory; such as those depending on equipment only available in the field (e.g. outdoor systems such as radio communications), or those related to real time and dynamic behaviour.

Moreover, when introducing moving block systems testing, we are regularly faced with the challenge of dealing with an explosion of the number of test cases due to the system operational complexity and dynamic characteristics, such as the case of calculation of movement authority under moving block systems.

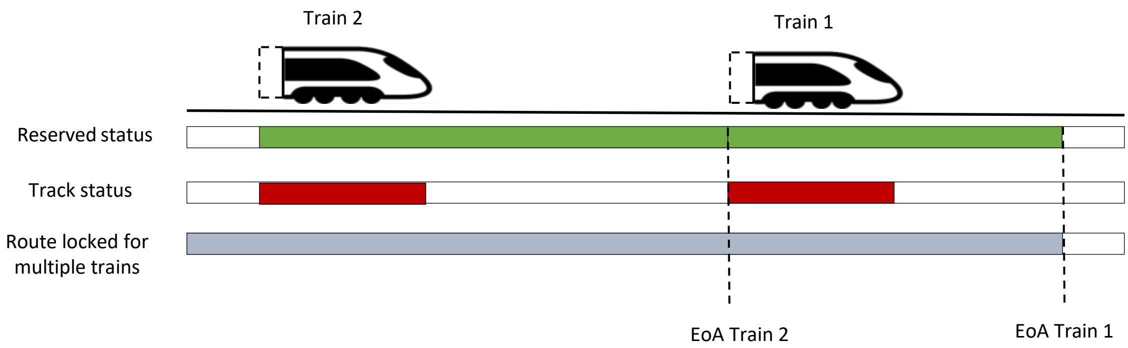
4.1 Overview of MA under moving block systems

While in ETCS levels 1 and 2 the movement authorities are determined using a combination of fixed blocks in order to allow trains to move on the track, under ETCS-L3, the blocks are no longer defined by pairs of adjacent fixed points on the line. Rather, their establishment is defined virtually, and movement authority is secured via radio-based communication.

In turn, in systems using a full moving block solution the MA can be issued to an arbitrary location, such as the rear of a preceding train. In a fixed block or virtual block system the MA shall be determined by boundaries of the fixed blocks [9].

The architecture presented in PART A allows movement authority tests in a laboratory environment to be addressed, thus saving cost and time while still providing the necessary information to verify the real behaviour of the system. Table 1 represents a description of a general MA under moving block and uses the concept of a route, which describes a path in the railway authorised for the passage of one or multiple trains, as defined in [9]

Table 1: Moving block movement authority allocation example

Movement authority
Scenario: Train 1 moves and RBC updates MA of Train 2 to the new rear of Train 1

Description: In this scenario a Full Moving Block system is considered. As Train 1 is moving away, there is the potential for the MA to be regularly updated for Train 2. As a result, a route must be locked, and the area reserved before the movement authority should be extended when applicable. For a full moving block system, the obstruction is marked as the rear end of the rightmost train in the MA generated by the RBC.

Due to the dynamic nature of moving block and virtual coupling, both solutions have the potential of significantly increasing the operational complexity by creating an infinite number of movement authority allocation scenarios in contrast to the current fixed block case. Consequently, the number of tests required to test every MA possibility increases exponentially, along with the effort required for a comprehensive test. Solely executing some approximation of all of these test cases would require a lot of time, but other related activities, such as the process to determine the appropriate test input and output and the analysis of the results of each test case would demand a major and probably infeasible amount of effort.

Currently, the most automated part of testing among the railway sector is the execution phase. The test strategy and approaches selected by the organisation will also play a key role and directly influence the testing effort requirements. Applying automation techniques in certain phases of the test process, such as creating and executing test cases, or even automated analysis of the test results instead of manual analysis of a huge set of data, can reduce the amount of testing time. Moreover, the use of metrics can help to predict potential defective areas and reduce the test effort. As defects are often not distributed equally, this technique can be implemented to support testing activities and reduce the need for each part of the system to be tested thoroughly.

Derived from those premises, approaches to support the test automation strategy have been organised into five key principles, as follows.

4.2 Test automation strategy

Some of the most common reasons to use automation in the testing process is to execute a set of tests faster, resulting in a reduction of costs and time in the testing process. In order to achieve this, it is very important to define the test automation process to fulfil the stakeholders' needs.

However, before choosing any idea of automation framework or script, the test automation strategy must be clearly specified.

According to [10], a test automation strategy can be defined as an action plan designed to provide broad guidelines so that the stakeholders can identify the suitable approach for design and implementation of the test automation. From a broader perspective, testing automation strategy derives from the definition of the stages illustrated in Figure 10.

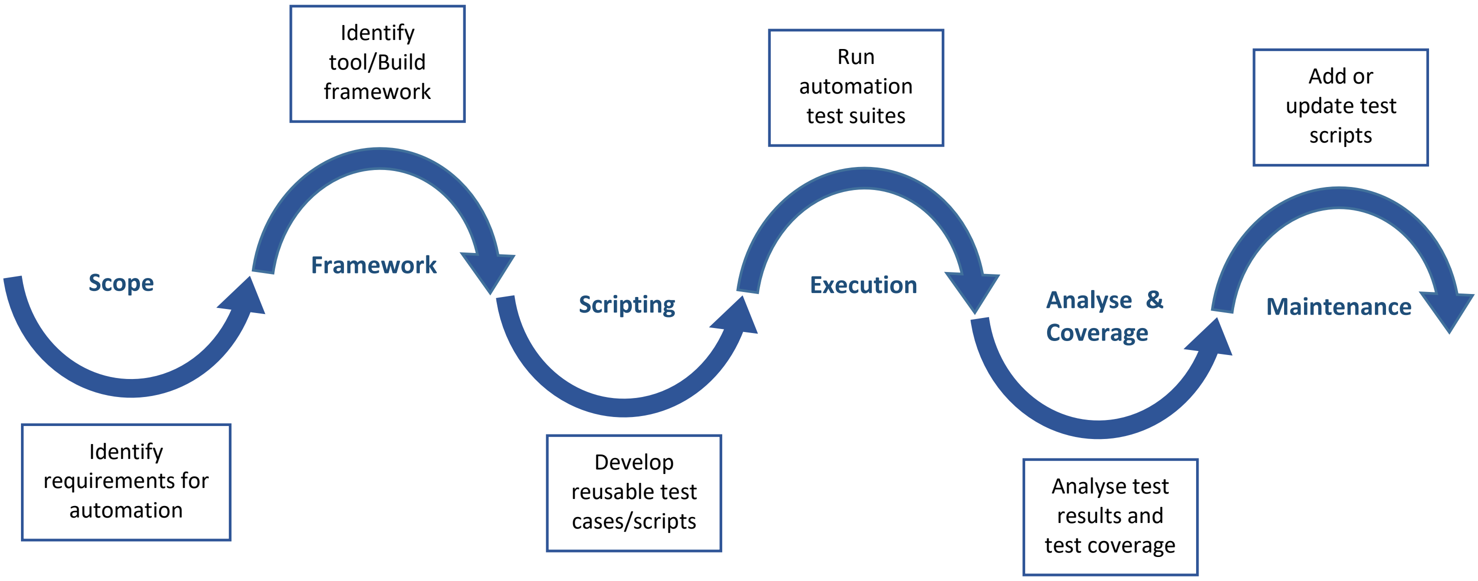


Figure 10: Test automation strategy

4.2.1 Scope identification

The first step in planning for test automation is to identify some key factors to support the definition of a concise strategy. Those factors and consequently the requirements for the test automation, will vary from organization to organization depending on:

i. The degree of automation desired

The ultimate goal of any type of test automation is to provide automated support for testing procedures. As a result, the degree of automation desired in the testing process defines the scope of automated testing. Possible levels of test automation can be described as follows:

Table 2: Levels test of automation

Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
No Autonomy	Assistance	Partial automation	Conditional automation	High automation	Full automation
Human is performing all test activities manually	Human control and monitoring with machine executing some subtasks.	Human now remains in the control loop and directs the machine.	Subtasks are control and perform by the human or the machine depending on the characteristic of the task.	Human has a supervisory role with machine performing all test activities.	Human has no role in the test activities.

ii. Organisation current state

Depending on the situation, which includes software, hardware, test cases, data and procedures currently in place for all the test activities, the organization can select a unified solution for the automated testing system or choose instead to implement multiple tools and mechanisms for different parts of the testing process or testing architecture.

The assessment of current state also has an important role to map the limitations of current practices and identify the key points that need to be addressed in order to achieve the level of automation desired.

iii. Functions that should be automated

It is also important to identify what it is outside the scope of the automation framework. Early decisions on which part of testing process is being automated, the components to include or exclude and which features are most important and should be automated or not, can collaborate to determine the strategy and minimize the complexity of the automation tasks.

iv. Test environment characteristics

When dealing with TCS, the requirements for automation can be very complex, particularly considering that the test bench is often reconfigurable to involve simulations or actual target equipment for some of its elements, and possibly may include equivalent items from different suppliers. The automation strategy must take into consideration that changes will occur and be designed to minimize the impact of such changes. Maintainability plays a key role in this process and will be addressed later on in Section 4.2.6.

4.2.2 Framework and design approach

A test automation framework outlines a set of guidelines, functions, concepts, processes and procedures utilised to support the design, creation and implementation of test automation.

Once the framework is created, it can be used with few changes in configuration, test data and object repository. It helps in fast execution with minimal human intervention. Most test automation frameworks support (i) the aim to evaluate expected results; (ii) the ability to share common test cases; and (iii) the parameter sets to easily organize and run tests.

In any type of test automation framework, we need the following elements [38][39]:

TEST CASE

"A set of input values, execution preconditions, expected results and execution post-conditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement" [43].

TEST SCRIPT

A test script is a set of instructions (written using a programming language) that is performed on a system under test to verify that the system performs as expected [41].

TEST DATA

Test data can be described as the set of inputs required to perform a test – or the data used to test specific functionalities of the application, such as user data, search queries, expected messages in case of invalid input, system parameters, etc.

LOCATORS

Locators are identifiers for the application elements such as buttons, alerts, etc.

The general correlation between the elements in a test automation framework can be seen in Figure 11 [40] below:

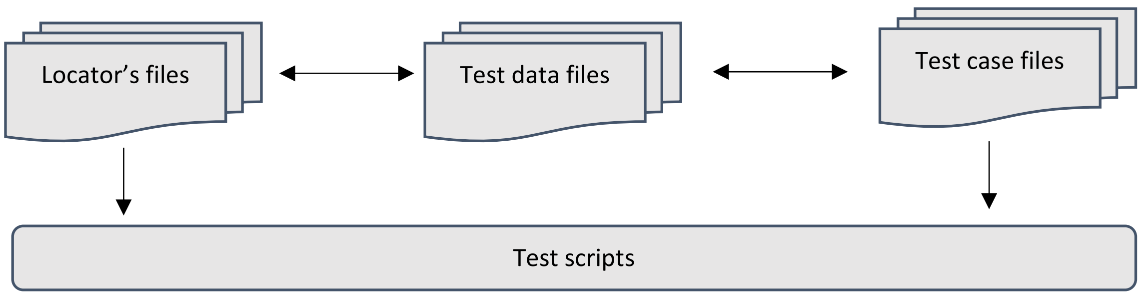


Figure 11: Test automation framework design

The difference between the various types of test automation framework designs is usually based on how and where the test case, test data and locators are defined and depending on the layer where the three elements above interact.

The overall automation framework selection and design are derived by the organizational goals, domains, business, and so on. The aspects that must be defined to select the more suitable framework type will vary from organization to organization. A few aspects to consider are as follows:

i. Increase reusability
ii. Enhance test coverage
iii. Speed up testing for multiple and frequent releases
iv. Endure consistency

v. Improve the reliability of testing
vi. Scalability
vii. Should have short driver script
viii. Maintainability

4.2.3 Script creation

As discussed in MOVINGRAIL Deliverable 2.1: *Moving block signalling test strategies system,* test cases normally are built as scenarios that describe each step and event and can be generated using different levels of automation.

One common goal of test automation is to reduce the number of test cases executed manually. Therefore, the process of creation of test scripts is an essential step and must conform to the test framework chosen. The list of goals for the scripts and code creation is based on the organization objectives and as such, the following list may vary:

i. The test should always consider a common start and end point
ii. The test should reveal maximum information in case an issue is found
iii. Appropriate comments and scripts should be presented
iv. The code should be readable and appropriate documentation should contain all the necessary information
v. The script should be maintainable and easy to modify
vi. Error handling and snapshots should be presented in the case of error
vii. Logs should be recorded and available at the end of execution
viii. Test Scripts, Test Cases and Data shall be formally cross reviewed (as per EN50128 and EN50129 standard).
ix. Choose the most suitable tool for automation

Figure 12 shows a general approach, where an example of organisation utilises the fact that the primary requirements defined for ETCS-L2 are still applicable for moving block systems - with some level of modification.

Considering that many ETCS-L2 test scenarios/scripts will already exist in libraries maintained by the train control supplier and/or the railway organisation, requirements for both L3 and L2 are compare and contrasted, in order to identify overlaps between them. After that, all existing test cases are divided in three parts: (i) Part A: Test cases that can be reused integrally without any changes (ii) Part B: Test cases that might require some level of modification to fit moving block requirements, and (iii) Part C: Test cases that do not apply to system requirements. In parallel, in the green branch, additional test cases are generated to satisfy and conform to the ETCS level 3 system requirements.

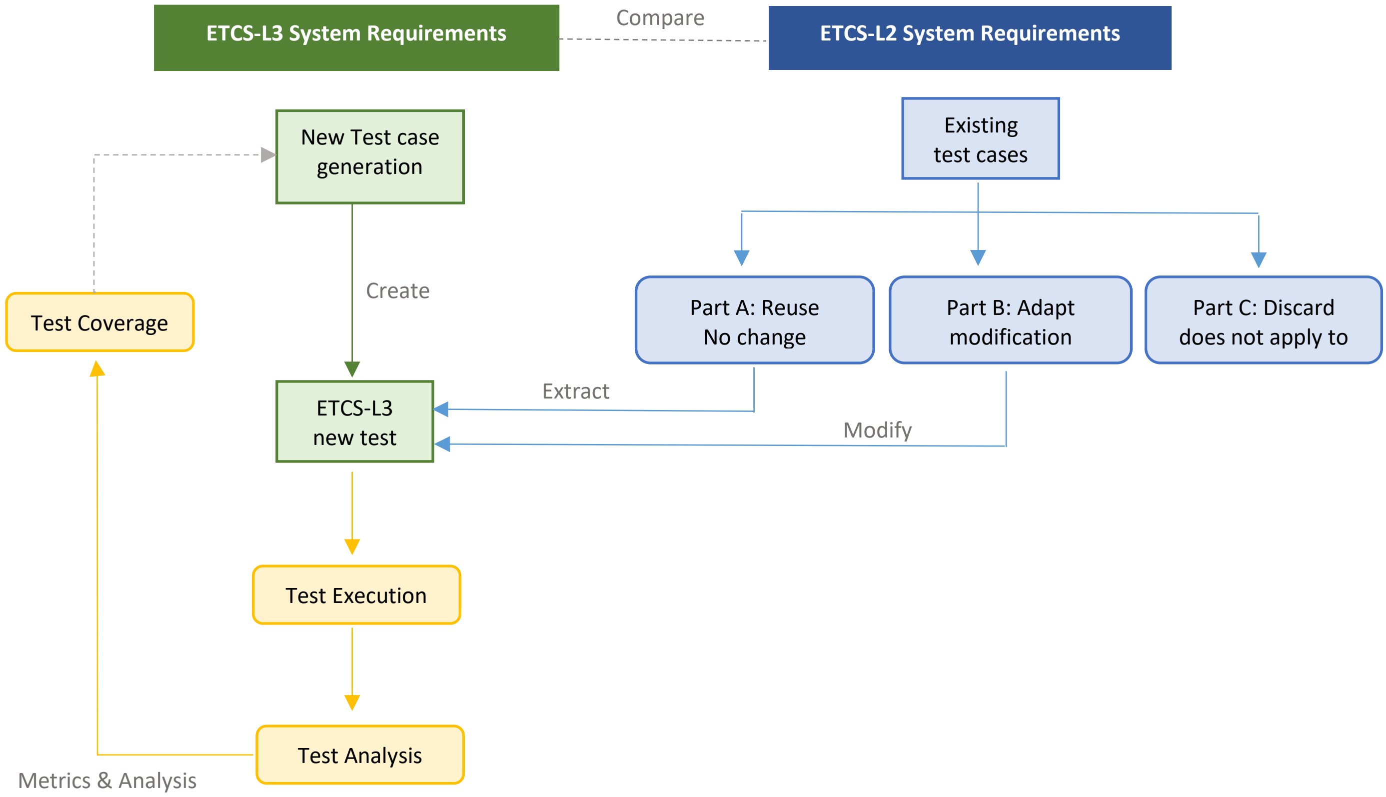


Figure 12: ETCS-L3 - Test case generation

4.2.4 Test execution

The level of automation adopted during the previous phases directly affects the test execution and, in this phase, the system under test is executed according to the framework selected and the test cases previously generated. The common actions performed in this phase can include:

i. Select a set of test scenarios to allow all the necessary behaviour to be checked
ii. Load test inputs to create each test scenario
iii. Run the test
iv. Log and capture the test results
v. Present the test results to be analyse

4.2.5 Test automation analysis and coverage

Essentially, test coverage helps monitor the quality of testing, and assists testers to create tests that cover critical areas that are missing or require validation. However, regardless of the method or tool chosen to create the scripts, maintaining and editing scripts might be very time consuming due to the endless need for creating or editing existing test scripts in order to cover more parts of the SUT.

In train control systems test coverage is limited as much by execution as by test script building and maintenance. In turn, generic test scripts can be used repeatedly (instantiated with individual test case data), since the same tests to check configuration data correctness and accuracy are needed at every instance of a particular scenario (e.g. a critical EoA), where there may be hundreds or

thousands of such critical EoAs in a single project.

In order to achieve the desired outcome, it is essential to set measurable goals and clear objectives for the test automation process, based on the test coverage and specific goals of each stakeholders and test plan. A few examples are:

i. Define the number or percentage of test cases to be automated
ii. Define the level of automation desired to the analyse of test result
iii. Identify applicable tools and approaches to validate test results
iv. Enhancing the test coverage by percentage
v. Reducing the time to release the SUT
vi. Reduction of test cycle time of new releases
vii. Ensuring relevant and up-to-date test data
viii. Conducting reviews and implementing metrics

4.2.5.1 Test automation metrics

Test automation metrics help to evaluate the effectiveness of the approach chosen, compare alternatives and monitor improvements and maintainability.

Implementing automated testing is long a process and any metrics chosen must take into account the organization’s needs. Some of the following attributes can be selected to measure the test automation strategy:

Efficiency: Efficiency is directly related to cost and can be measured based on the elapsed time (e.g. hours elapsed) or effort (working hours) required to perform a specific task.

Maintainability: Maintainability is measured based on the average elapsed time or effort required to update a test script.

Flexibility: Flexibility is related to the extent to which the test cases can be combined in order to achieve different test objectives. It can consider different parameters, such as the time taken to identify the required test cases for a specific purpose, or the effort needed to restore a deleted test case.

Reliability: Reliability is linked to the test’s ability to provide accurate and repeatable results. Scales of measurement can be based on different parameters, such as the number of false positives and false negatives, or the number of test cycles required due test defects.

Robustness: Robustness is related to the ability to cope with unexpected issues during the test execution, the frequency of failure and the time taken to investigate the causes that resulted in the test failure.

Usability: Usability is defined in terms of the time taken to add new test scripts to an existing test suite, the training time needed to utilise the test suites and how easy it is for the user to execute and maintain the test automation process.

Portability: Portability is measured based on the time or effort required to run a set of test scripts in different conditions (e.g. test environment updates or new test tools).

4.2.6 Maintainability

According to [41], on average, 25% of an application is rewritten each year. As a result, if the test scripts, related to the application, cannot be changed in a reasonable amount of time, the whole test automation process becomes obsolete. Therefore, maintainability is a key aspect to consider, not only to ensure that changes can be easily identified and made, but also to prevent any unexpected impact or consequence due poor test design or implementation.

In order to apply maintainability to test scripts, the stakeholder's goals and characteristics must be taken into consideration. This list will vary from organization to organization, but a few examples can be seen below:

i. Business requirement complexity
ii. Application complexity
iii. Data structures complexity
iv. Code complexity
v. Change history documentation
vi. Automated documentation
vii. Business overview documentation
viii. Test script annotation
ix. Test framework
x. Release frequency
xi. Test data format

4.3 Summary of moving block system test automation

This section discussed how the architecture proposed in the previous section requires a well-defined test automation strategy in order to overcome additional challenges of testing moving block systems, such as the case of explosion of test case in scenarios related to movement authority.

As mentioned, the test automation strategy adopted will vary according to the organization and the current procedures in place. However, considering that nowadays the most automated part of testing among the railway sector is the execution phase, a set of high-level approaches were presented to support the definition of an automated testing strategy to test moving block systems and minimise on-site testing.

5. Conclusions

The testing of train control systems is essential to ensure their reliable and safe performance. Testing systems and strategies for ETCS level 2 are in place, which use combinations of simulations to emulate the real railway, bespoke testing software within architectures that allow the hardware or software under test to be placed in the testing loop. As moving block / ETCS level 3 systems are specified, the testing systems and strategies must be appropriately extended. This deliverable, in combination with MOVINGRAIL D2.1 *Moving Block Signalling System Testing Strategy* defines improved strategies and methods for the functional and data testing of moving block signalling systems for future engineering projects. This has been approached by identification of the functional differences between ETCS level 2 and ETCS level 3 systems, and by consideration of scenarios and methods that allow a move as much as possible away from on-site testing.

Following the operational concept and higher level testing strategies described in D2.1, this has been done in two ways: the specification of a testing architecture and its interfaces for a moving block system, and a description of testing strategies and metrics for measuring the quality of testing. The X2Rail suite of projects has been called upon to provide the requirements and specifications for moving block, and allow identification of differences from ETCS level 2 systems. In the moving block paradigm, movement authorities can be allocated to any point on the track and move with trains, meaning an expansion to the number of scenarios that must be tested. The positioning and train integrity are managed on-board rather than from the trackside and although the communication systems provide the same functionality, the architecture of both the on-board and trackside components differ between ETCS level 2 and ETCS level 3. Cases where communication is lost must also be managed differently. These differences are taken into account in specifying an architecture appropriate for moving block operation, and in the interfaces within the testing components. A moving block movement authority allocation scenario is discussed in the context of the proposed testing architecture to highlight the important issues relating to test automation, such as apportionment of effort and testing time. Following a structured test plan and increasing automation where appropriate should allow testing efficiency even in the more complex case of moving block and its expanded movement allocation possibilities.

References

[1] *Introduction to Railway Signalling, IRSE, 2014, ISBN 978-0-902390-30-0*. Institution of Railway Signal Engineers.

[2] 'White Paper European Railway Traffic Management System - ERTMS'. http://www.climateaction.org/images/uploads/documents/European_Railway_Traffic_Management_System.pdf (accessed Jan. 30, 2019).

[3] 'X2RAIL-1'. https://projects.shift2rail.org/s2r_ip2_n.aspx?p=X2RAIL-1 (accessed Jun. 04, 2020).

[4] 'X2RAIL-3'. https://projects.shift2rail.org/s2r_ip2_n.aspx?p=X2RAIL-3 (accessed Jun. 04, 2020).

[5] 'X2RAIL-2, D3.1, "System Requirement Specification of the Fail-Safe Train Positioning Functional Block", 18/12/2018.' .

[6] 'X2Rail-2, Deliverable D4.1, Train Integrity Concept and Functional Requirement Specification, 2017'. .

[7] 'X2Rail-2, Deliverable D6.3, Description of Use-Cases for new TMS Principles, 2018.' .

[8] 'X2Rail-1, Deliverable D3.3, Specification of the communication system and Guideline for choice of Technology, 2017.' .

[9] 'X2Rail-2, Deliverable D5.1, "Moving Block System Specification", 2019'.

[10] X2Rail-1, Deliverable D6.1 Current test condition and Benchmarking report, 2017.

[11] N. Furness and M. Bartholomeus, 'ERTMS Level 3: the Game-Changer', p. 8.

[12] 'Hybrid ERTMS/ETCS Level 3, ref: 16E042, version 1A, 2017 by EEIG ERTMS Users Group'. http://www.ertms.be/sites/default/files/2018-03/16E0421A_HL3.pdf (accessed Jun. 30, 2020).

[13] 'X2Rail-1, Deliverable D3.1, User & System Requirements (Telecommunications), 2017.' .

[14] UNISIG Subset-037: EuroRadio FIS, rev 3.2.0, 17/12/2015.

[16] 'TSI Commission regulation (EU) 2016/919, of 27 May 2016, on the technical specification for interoperability relating to the "control-command and signalling" subsystems of the rail system in the European Union Set of specifications # 3 (ETCS baseline 3 and GSM-R baseline 1)', p. 79.

[17] 'X2Rail-2, Deliverable D5.2 Moving Block Operational and Engineering Rules, 2019.' .

[18] J. S. Keränen and T. D. Räty, 'Model-based testing of embedded systems in hardware in the loop environment', *IET Software*, vol. 6, no. 4, p. 364, 2012, doi: 10.1049/iet-sen.2011.0111.

[19] Gonzalo Solasa,*, Jaizki Mendizabala, Leonardo Valdiviaa, Javier Añorgaa, Iñigo Adina, Adam Podhorskia, Stanislas Pinteb, Luis Gerardo Marcosb, Jesús Ma Gonzálezc, Francisco Cosínc, 'Development of an advanced laboratory for ETCS applications', Feb. 04, 2019. https://www.usenix.net/legacy/publications/library/proceedings/lisa94/full_papers/eirich.ps (accessed Feb. 04, 2019).

[20] MOVINGRAIL, Deliverable D2.1, 'Moving block signalling system test strategy, 2019

[21] UNISIG SUBSET-026-2: System Requirements Specification rev 3.6.0, 13/05/2016

[22] UNISIG SUBSET-94: Functional requirements for an onboard reference test facility, rev 3.1.0, 24/04/2017.

[23] UNISIG SUBSET-111: IOP Test Environment Definition UNISIG, rev 3.5.0, 17/02/2016.

[24] UNISIG SUBSET-076-6-3: Test Sequences, rev 3.2.0, 30/05/2017

[25] UNISIG SUBSET-041, 'Performance requirements for interoperability' rev 3.2.0, 17/12/2015

[26] UNISIG SUBSET-026: System Requirements Specification, rev 3.6.0, 25/05/2016

[27] In2Rail, Deliverable D7.1, State-of-the-Art and High Level Requirements, 2016.

[28] 'UNISIG SUBSET-039: FIS for the RBC/RBC Handover, v3.2.0, 17/12/2015.' .

[29] UNISIG SUBSET-111-3: Interoperability Test Environment Definition (FFFIS for TCL-RBC adaptor), rev 3.5.0, 17/02/2016

[30] 'Digital Railway – ETCS Onboard System Definition'. http://digitalrailway.co.uk/wp-content/uploads/2019/04/153821-NWR-REP-ESE-000005-ETCS-Onboard-SDD_3.0.pdf (accessed Apr. 22, 2020).

[31] UNISIG Subset-034: Train Interface FIS, rev 3.2.0, 17/12/2015

[32] UNISIG SUBSET-058: Application Layer STM FFFIS, rev 3.1.0, 16/12/2015

[33] UNISIG SUBSET-056: Safe Time Layer STM FFFIS, rev 3.0.0, 29/02/2012

[34] UNISIG SUBSET-057: Safe Link Layer STM FFFIS, rev 3.1.0, 16/12/2015

[35] A11 T6001, Radio transmission FFFIS for Euroradio, rev 13.0.0, 18/12/2015

[36] UNISIG SUBSET-036: FFFIS FOR Eurobalise, rev 3.1.0, 17/12/2015

[37] UNISIG SUBSET-044: FFFIS for Euroloop, rev 2.4.0, 29/02/2012

[38] P. Ammann and J. Offutt, 'Introduction to Software Testing', *Cambridge Core*, Dec. 2016. /core/books/introduction-to-software-testing/95E57CCADEA697EC8594F03729F47311 (accessed May 18, 2020).

[39] S. Amaricai and R. Constantinescu, 'Designing a Software Test Automation Framework', *Informatica Economica; Bucharest*, vol. 18, no. 1, pp. 152–161, 2014, Accessed: May 18, 2020. [Online]. Available: http://search.proquest.com/docview/1525467700/abstract/5AFE3708BD37419FPQ/1.

[40] A. Bhargava, *Designing and implementing test automation frameworks with QTP / Ashish Bhargava.* Birmingham: Packt Publishing, 2013.

[41] 'Automated Testing Handbook'. http://www.softwaretestpro.com/wp-content/uploads/2016/10/AutomatedTestingHandbook.pdf (accessed May 18, 2020).