A Flexible Simulation Environment for Railway Traffic Management Systems

Francesco Flammini and Pasquale di Tommaso

ANSALDO SIGNAL – Ansaldo Segnalamento Ferroviario S.p.A., Naples, Italy

Summary
ERTMS/ETCS is going to become the reference standard for modern railway signalling. To develop a safe and reliable Automatic Train Protection System (ATPS) based on ERTMS/ETCS, a detailed functional testing phase is needed, meeting the requirements of international railway safety standards. In this paper we deal with the functional validation of the trackside part of an ERTMS/ETCS compliant system. An extensive set of functional tests have been specified in order to thoroughly verify the system, using an innovative approach based on influence variables and state diagrams. However, such a detailed test specification requires a great amount of time and resources to be entirely executed in the real environment. Moreover, several tests need to generate abnormal safety-critical conditions that are unfeasible on the field. In this paper we describe how we overcame such problems using a specific simulation environment capable to quickly and automatically execute anomaly tests in normal as well as in degraded operating conditions.

Keywords:
ERTMS/ETCS, Safety-Critical Systems, Verification & Validation, Functional Testing, Simulation Environments

1. Introduction

The aim of our work was the execution of the functional tests specified for the AV railway signalling system (AV is the acronym of "Alta Velocità", which is the Italian for "High Speed"). The AV signalling system is based on the European Railway Traffic Management System / European Train Control System (ERTMS/ETCS) Level 2 trackside (i.e. ground) sub-system specification [1] and is adopted in Italy in the new developed high speed railway lines. Being a safety critical railway control system, ERTMS/ETCS needs a thorough testing activity in order to be completely validated. General requirements on the validation process are provided by international safety and reliability standards [2] and they stress the importance of functional testing as one of the most important steps in ensuring system safety. Functional testing, usually based on a black-box scheme, is aimed at verifying system implementation against its functional requirements. The most important aspect in our case was the verification of interoperability and safety requirements. Compliance to ERTMS/ETCS, in fact, means also interoperability of trans-European rail lines. Safety, of course, was the most important aspect: all the functional safety requirements, obtained by a preliminary hazard analysis process [5], were to be thoroughly verified. This implied a detailed functional test specification, based on the concepts of influence variables, firstly introduced in the SCMT system validation (see [3, 4]), and state diagrams, found to be the best way to represent the behavioural aspects of a very complex system, as the one under test. In the total scheme of the assurance tasks (hazard analysis, static code analysis, etc.), functional testing plays, according to our experience, the most important role, in terms of required time, budget and criticality (it is one of the last activities to be performed before system activation).

The main problem was that such a thorough test specification included more than 2000 tests, many of which were not reproducible in real conditions, as they regarded extensive combinations of abnormal conditions, negative inputs, degraded states of operations, etc. Thus, the testing team had to deal with the following three issues:

- a lot of test conditions (about the 30%) were not feasible in the real environment;
- the time to execute the tests in the real environment was excessive (it would have taken several years);
- the real environment does not allow to automate test execution, and this is a serious problem when dealing with regression testing (the entire test suite must be repeated at any new software version).

Therefore, a simulation environment had to be developed and fine tuned to match the needs of the test engineers, consisting in simulating both nominal and negative test conditions, also in degraded states of system operation. Finally, the simulation tools had to be able to support batch execution by means of proper script management capabilities, in order to automate the test process. The "system in the loop" [8] simulation environment described in this paper together with a specifically designed anomaly management tool allowed the testing team to define by script files and automatically execute most of the specified functional tests in a few months, detecting several unconformities and

Manuscript received October 5, 2025
Manuscript revised October 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.10.16

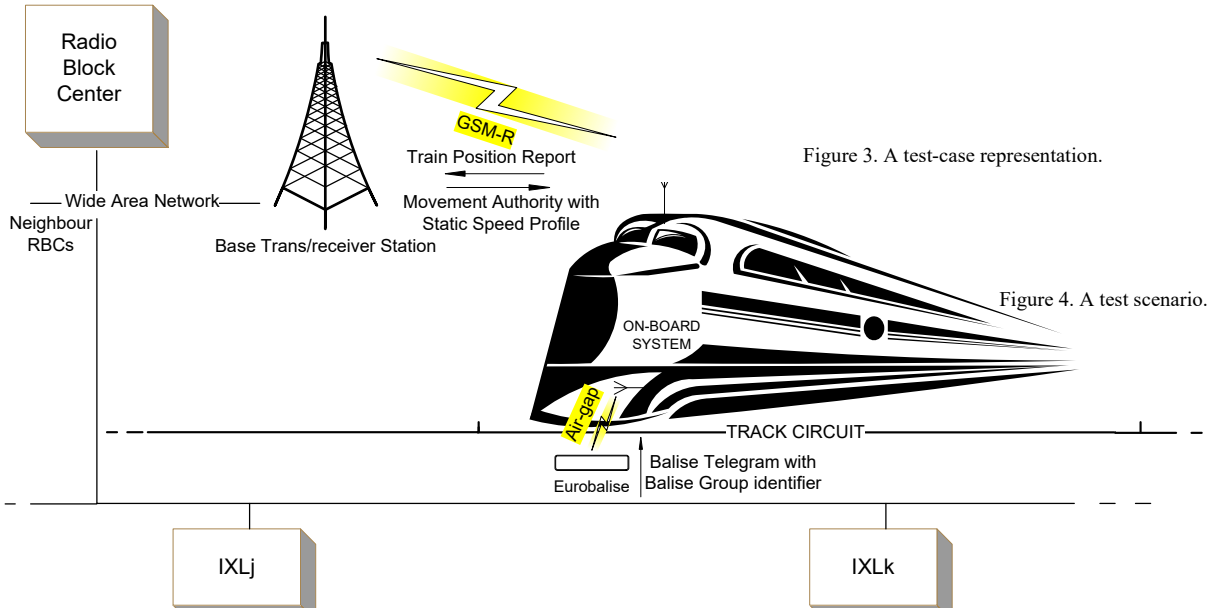


Figure 3. A test-case representation.

Figure 4. A test scenario.

Figure 1 An ERTMS/ETCS Level 2 system.

implementation errors (test suite execution is still in progress).

This paper is organised as follows: Section 2 describes the system under test, in terms of working principles and hardware architecture; Section 3 presents the background and an overview of the methodology employed for test specification; Section 4 provides a description of the logic and hardware structure of the simulation environment used to execute the specified tests in a quick and fully automated way; Section 5 concentrates on the management of anomalies, that is abnormal test conditions that are generated by a proper ad hoc tool that integrates the nominal simulators; Section 6 shows some examples of tests whose execution would be very difficult or impossible without the trackside simulation environment integrated with the anomaly manager tool; finally, Section 7 contains some closing remarks and a brief discussion about the future applications of the simulation environment presented in this paper.

2. The ERTMS/ETCS trackside system

ERTMS/ETCS is the specification of a standard aiming at the improvement of safety, performance and interoperability of European railways. In Italy, the so called Level 2 specification of ERTMS/ETCS is used on high-speed railway lines.

The trackside subsystem is the “ground” (or “fixed”) part of the overall signalling system, that is the entire ERTMS/ETCS system minus the on-board sub-system. In the ERTMS/ETCS specification, system architecture includes the following main sub-systems: on-board, which performs train protection by controlling train speed against the elaborated dynamic speed profiles; trackside, which collects track and train information in order to feed the trains with the needed data (i.e. the distance they are allowed to cover and the speed restrictions); the lineside, which includes the balises (or Eurobalises, as defined in ERTMS), that are devices positioned along the track that transmit static information to the on-board sub-system. In ERTMS L2 balises act just like milestones to allow trains to detect their position and communicate it to the trackside sub-system, which will use such information to provide train separation.

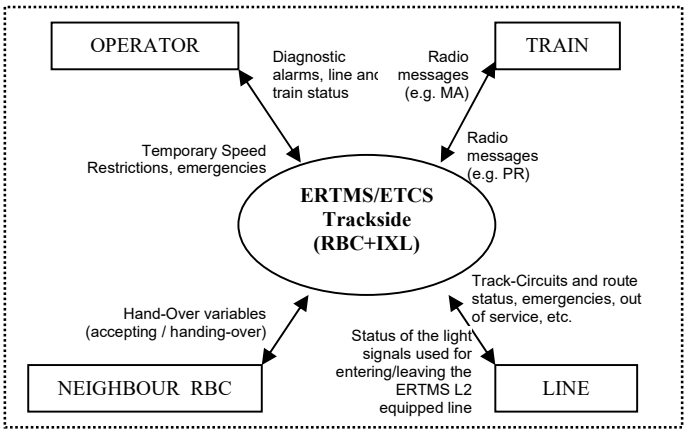


Figure 2 The trackside context diagram.

At level 2 the latter two subsystems (trackside and

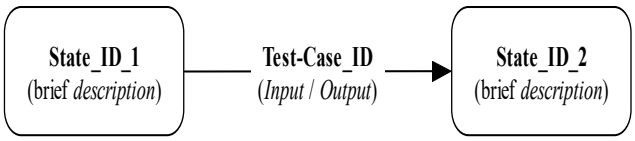
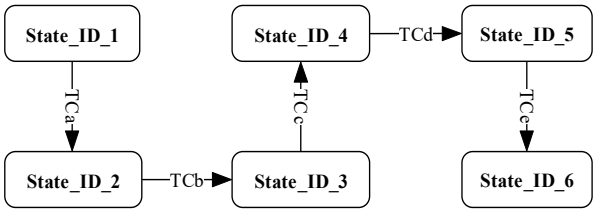


Figure. 3 A Test Case ID .

lineside) can be joined together to form the "routing and separation" subsystem. Therefore, in the following by referring to the trackside sub-system we will also include fixed balises ("fixed" means they can only transmit unchanged, static information).

The ERTMS L2 trackside subsystem, as it will be explained in depth in the following of the paper, is mainly constituted by two sub-systems: the route management system (known as Interlocking, or IXL), which is responsible of train routing and of collecting track circuit occupation status, and the separation subsystem, made up by the Radio Block Center (RBC) and Eurobalises, which is mainly responsible of detecting train position and delivering the correct Movement Authorities (MA) and Static Speed Profiles (SSP) to the trains. The IXL part has not been standardized in the ERTMS/ETCS specification, so it was possible to simply adapt the already existing national interlocking system. The Italian national IXL is a distributed system, made up by a series of distributed IXL modules (hence indicated with IXL1, IXL2, and so on) connected to each RBC in order to detect and transmit route and track status to the separation sub-system.

The lineside sub-system is made-up by Eurobalises, which transmit a position telegram when energized by a train passing over them. Such a telegram contains a Balise Group (BG) identifier that will be included in the train Position Report (PR), together with other information (e.g. train speed and position detected by the on-board odometer), and transmitted to the RBC. RBC will use the balise identifier included in the PR and the offset position measured by train odometer in order to calculate the Movement Authority to be sent to the train. In fact, RBC has an internal data-base (configured off-line), in which BGs are associated with their actual position and with Static Speed Profiles (SSP). This information, together with the track circuit status received from the interlocking system is (nearly) all the RBC needs to continuously provide trains with their MAs and thus to achieve its separation functionality. In ERTMS L2 the on-board and trackside communicate by the GSM-R radio network, especially designed for railway applications, using the Euroradio protocol [10]. Data is encapsulated in radio messages whose type and structure is standardised in the ERTMS/ETCS specification.



TSi = TCa + TCb + TCd + TCe

Figure. 4 A logic scheme of the testing environment.

The overall system architecture and the main data flows are depicted in Figure 1, while a context diagram of the trackside subsystem is reported in Figure 2.

3. The Test Specification Methodology

Traditional functional testing techniques allow to verify system implementation against its requirements, but it is beyond their scope to validate system requirements specification; natural language specification, however, even though revised, is often incomplete, so a stronger technique is needed. This technique should merge the main objectives of safety and feasibility. The test specification for the ERTMS/ETCS trackside system had to guarantee:

- The complete coverage of system functional requirements, both in nominal and degraded states of operation ("negative testing");
- An in depth analysis of system scenarios aimed at detecting operating conditions not covered by system specification (using the concept of "influence variables", as described in [3, 4]);
- The minimization of the number of required test-cases, to ensure the feasibility of the functional testing phase;
- A structured and systematic test specification, documentation and execution process, aimed at an easier data understanding and management, to be shared by a large group of test engineers.

The result of considering all these needs was a test specification methodology based on influence variables and represented by state diagrams. The influence variables are all system variables that are able to influence its behaviour and have been divided in input and state variables. The resulting state diagrams represented all the system operating scenarios that are ideally linkable all together to represent the overall functional behaviour of the system under test. The test specification process is made up by the following steps:

- Detection of system boundaries, to highlight input-output gates;
- Elaboration of a list of base operational scenarios, to be used as a starting point for the functional analysis;
- For each scenario, detection and reduction of influence variables (system level variables, obtained by the specification, influencing system behaviour);
- For each scenario, representation of system behaviour in the functional scenario by means of a state diagram;
- For each state, generation of the elementary test-cases (simple "input-output-next state" relations);
- Generation of scenario test-cases, by linking elementary test-cases.

More specifically, the following were the significant state variables for the trackside system:

- Track status (managed by IXL): track circuit occupation (used to compute the Movement Authority); route integrity (e.g. "switches out of control"); emergency conditions (e.g. "line out of service").
- Train status (as seen from the RBC): information received by means of the train Position Report (train speed and position, as computed by the odometer, and Last Relevant Balise Group read by the train); information previously managed by the RBC (Movement Authority and SSP assigned to the train, list of messages waiting for an acknowledgement, list of emergency messages transmitted to the train).
- RBC status: list of radio messages sent to the train; list of radio messages received from the train; route status (i.e. route assigned to the trains); emergency and Temporary Speed Restrictions input from operator.

Analogously we could define the input from trains (i.e. radio messages), from track (e.g. track circuit occupation) and from operator (e.g. Temporary Speed Restrictions on the line), which have to be managed from the trackside in any state. As for the expected trackside behaviour, most of the outputs directly regard RBC, which is the most complex and important subsystem, because it collects data from the track and directly interacts with the on-board subsystems. Generally speaking, there are some common aspects in RBC reaction against a particular input, which we briefly list in the following:

- When it receives an emergency condition from the IXL or from the human operator, it reacts sending a proper emergency message to one or more trains;
- When it receives a Position Report from a train, it stores the relevant information and verify the possibility to assign it a new Movement Authority;
- According to the track freedom and route integrity received from the IXL, it chooses the length and the operating mode (Full Supervision, On-Sight or Staff Responsible) of the Movement Authority to be sent to the trains;
- When actuating a procedure, that is a sequence of predefined operations, it ignores a set of "safe" unexpected messages received from a train while it orders a disconnection if the message is considered "unsafe";
- During a procedure, it passes from a state to another when it receives an expected relevant message from a train or a condition from the trackside;
- It manages some sets of messages at any phase of train mission (i.e. during any procedure), such as disconnection requests and validated train data.

The combination of a system state, a relevant input condition, an expected output and state transition constitutes an elementary Test Case for the system (see Figure 3), while several Test Cases linked together in order to reproduce a complete evolution of the system under test in a given scenario is named a Test Scenario (see Figure 4).

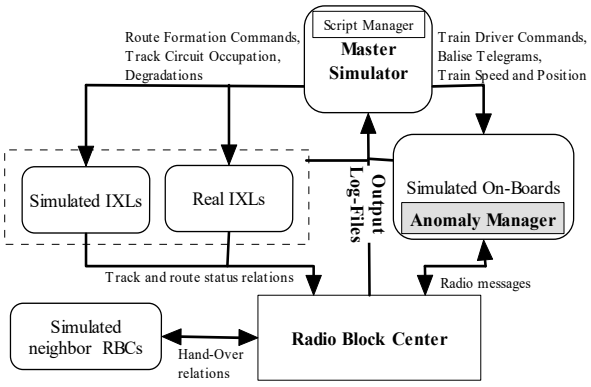


Fig. 5 A logic scheme of the testing environment.

4. The Simulation Environment

The IXL modules (see Figure 1) are distributed all along the track, at an average inter-distance of 12 km and are normally remote controlled by a central control room. There are several types of IXL modules, that we distinguish here only into "line" and "station" categories for the sake of simplicity. The Radio Block Center, instead, is physically installed in the central place and communicates with neighbour RBCs and its IXL modules by means of a high-speed long-distance fiber optic backbone (redundant). Given the complex architecture of the system under test, it was not easy to create a simulation

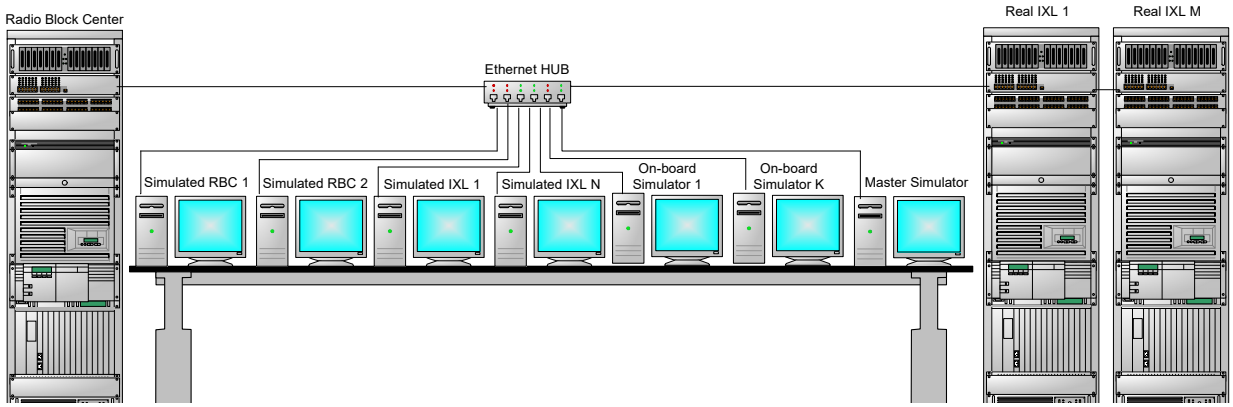


Fig. 6 The hardware structure of the trackside simulation environment.

environment that was both realistic (all the real hardware and software to be tested had to be used) and flexible (the external environment had to be completely programmable). A classic "system in the loop" scheme was adopted for the simulation environment; however, the hard task was to adapt the tools used to stimulate the system in normal operating conditions, that were already developed, in order to make them able to be used to generate non nominal (or negative) ones.

The simulation environment is made up by the following elements (see Figure 5):

- a real RBC;
- a pair of simulated RBCs (its neighbours);
- a certain number of real and simulated IXL modules (the ones in its supervised area);
- a certain number of on-board simulators, used as a sort of input injectors and output probes with respect to the trackside;
- a so called master simulator, used to control and stimulate all the simulated entities.

The choice of real and simulated sub-systems is given by two contrasting factors:

- the realism of the environment, which would suggest the use of all real sub-systems;
- the flexibility of the environment: most negative and degraded conditions are either impossible or very difficult to obtain with the real systems.

The master simulator is a tool used to command the on-board and IXL, in particular by stimulating:

- the on-board simulator with train-driver, balise and trainborne (speed, diagnostic) inputs;
- the IXL with track conditions (track circuit occupation, degradations of route status, emergencies).

The configuration of the real RBC was based on the same hardware and software used on the field, comprising a vital section constituted by the following parts:

- a safety kernel with three independent computing subsystem in a Triple Modular Redundant configuration, which manages the train separation;
- a Man Machine Interface (MMI) constituted by a video terminal (showing train data), a track display (showing train position and track status) and a functional keyboard used by the RBC operator to digit commands to be sent to the trains (e.g. temporary speed restrictions and emergency messages).

The non vital section is made up by the following systems:

- two communication computers, used to communicate with IXL and on-board sub-systems;
- a redundant chronological event recorder, that is a sort of extended Juridical Recording Unit (JRU) [1] with the aim of recording the times and nature of the significant events (e.g. diagnostic data, alarms).

The extended JRU is also used to read after-test data in order to compare obtained outputs with the expected ones (such comparison is partly automated). All the real and simulated entities were connected in our laboratory by means of a normal LAN (Ethernet 100Mb/s).

Each of the simulated sub-systems (i.e. each simulated on-board, RBC, or IXL module) was installed on a different general purpose computer (see Figure 6). The software simulators were complete, showing all the features of a real MMI on the PC screen. The master simulator together with the train simulators allows to simulate the trains marching on a certain track section, showing their speed and positions and simulating track circuit and route occupation.

In order to allow test automation, both in the real systems and in the simulated ones it was installed some tools which communicate with a central master by which it is possible to command all the systems and create all the abnormal and degraded conditions which could happen in a real operating situation. A scripting language was implemented in order to specify batch sequences, that is the commands of the master simulator used to execute the complete test scenarios.

The last element of the simulation environment is the on-board simulator. It has been developed to simulate the behaviour of a real train and properly adapted in order to generate anomalies that, otherwise, would be very hard or impossible to obtain with a real train. The simulation of anomalies with the on-board simulator is the main topic of the next section.

5. Simulation of Anomalies

The communication between the RBC and the on-board and between the RBC and the IXL uses an open network. The CENELEC 50159-part 2 [11] norms report the threats of a communication based on an open network (i.e. deletion, re-sequencing, insertion, repetition, delay, corruption and authentication of a message; see [11]) and suggests some means to ensure the safety of the system with respect to such threats. The communication protocol employed for the data exchange from and to the RBC is CENELEC compliant and should protect from all the aforementioned threats [9]. The functional analysis used for test specification had to consider all the possible threats also in degraded operating conditions (e.g. a degraded route due to a loss of control of one or more switches) in order to exercise the robustness of the systems and of the communication protocol.

Therefore, the simulation environment has to be able to simulate both degradations and malfunctions. The "standard" environment is able to create all the degradations of the signalling system, which are abnormal railway conditions which the system must be able to properly manage. The simulation of railway degradations is useful to verify that the RBC is able to understand and react correctly to such conditions, ensuring the safety of train movement.

However, the standard environment is not able to reproduce the anomalies of the communication between the RBC and the on-board, which can happen in real operating conditions. In fact, the same threats reported in the CENELEC 50129 part 2 can affect the communication by the GSM-R network and both the robustness and the protection mechanisms implemented at different levels (protocol, application, etc.) must be verified in the functional testing phase.

Table 1:: Threats of system communications.

CENELEC EN 50159 Keywords	
Keyword	**Meaning**
Repetition	A message is received more than once
Deletion	A message is removed from a message stream
Insertion	A new message is implanted in the message stream
Resequencing	Messages are received in an unexpected sequence
Corruption	The information contained in a message is changed, casually or not
Delay	Messages are received at a time later than intended
Masquerade	A non-authentic message is designed thus to appear to be authentic (an authentic message means a valid message in which the information is certificated as originated from an authenticated data source)

For the above considerations and due to the need for testing the train separation system in all conditions with all combinations of its significant inputs (as explained when describing the test specification methodology; see Section 1.2), the simulation environment had to be adapted and customized in order to simulate in laboratory all the aforementioned communication anomalies.

The first step was the analysis of test specifications in order to identify the tests related to communication anomalies. We found out that only the 13% of the specified tests corresponded to nominal operating conditions; the remaining 87% were degradation or anomaly tests (respectively the 52% and the 35%). On the basis of such analysis and classification, we implemented a so called "anomaly manager" tool, which was completely independent from the nominal simulator.

The abnormal conditions that have been detected and implemented are the following:

- deletion of any message from a train to the RBC;
- deletion of any message sequence (i.e. loss of N consecutive messages);
- substitution of any message with any other one;
- insertion of a certain massage in any correct message sequence;
- modification of one or more fields in any message to be sent to the RBC with erroneous values;
- one or more repetitions of a message.

The implementation of the anomaly manager, moreover, allowed to apply one or more abnormal

conditions in any phase of the train mission, depending on train position and on the message the train would send in nominal operating conditions.

The abnormal conditions are listed in a configuration file which the on-board simulator reads at the beginning of each test. This allows to automatically execute more consecutive tests comprising several abnormal conditions.

Before starting test execution it is necessary a preparation phase in which such configuration files for the on-board simulator must be compiled. Then the master simulator scripts must be prepared, and this allows to automatically execute, by means of a single key pressure, any sequence of complex test scenarios.

The described implementation of the anomaly manager allows to test the behaviour of the trackside system with all the inputs coming from the on-board sub-system. This, together with the already existing possibility of generating all the railway degradations, allows to execute any extensive test-set.

The overall simulation environment, comprising the anomaly manager, features several advantages with respect to the "on the field" execution (by means of real train runs), as we already mentioned, which reflects to the possibility to thoroughly verify the system under test in less time and at a less cost.

6. Anomaly Testing Examples

In this section we present some examples of application of the anomaly manager in the simulation of abnormal operating conditions. As already mentioned in Section 1.2, the Radio Block Center is designed in order to tolerate unexpected messages, by ignoring them or by automatically reaching a safe state (i.e. sending a disconnection request) whenever it detects a safety critical condition or a non Unisig compliant[1] on-board system. For each state of the functional test scenarios, the RBC is tested against all the expected and unexpected messages received from the on-board. All the non nominal conditions can only be tested by means of the Anomaly Manager, because there is no way to reproduce them with a real on-board system. Moreover, the RBC features some robustness against availability critical situations: for instance, if it does not receive a message from a certain train for a given time (a few minutes), it has to delete such train from its internal database; this situation corresponds to a "lost" train (not properly disconnected), which must not be managed by the RBC anymore. Then it is necessary to allow other trains, within the maximum number allowed by the RBC, to connect using the so freed channel.

More specifically, in the following we present three test-case examples that can be easily reproduced in the simulation environment by means of the Anomaly Manager:

a. Unknown balise group;
b. Unexpected train data;
c. Unauthorized Track Ahead Free (TAF).

The case (a) corresponds to a RBC receiving a balise group identifier (used as a location reference) which is not included in its database. This can be the result of several faults: a mis-positioning of the balises, a train connected to the wrong RBC, a configuration error in the balise telegram or in the RBC database (wrong ID or balise not configured at all). When the RBC receives a Position Report from a train with an unknown balise group, then it must not control the train because it does not seem to belong to its supervised area. Thus, the robust reaction which has been designed for the RBC against such a condition is the sending of a disconnection request to the train. With a real train, such a test would require a very difficult preparation (e.g. balise reprogramming or reposition). Using the Anomaly Manager, instead, it is sufficient to load the configuration files of the on-board simulator in order to substitute the message corresponding to a correct Position Report with one in which the balise group identifier is altered as requested by the test-case (i.e. set of a wrong number).

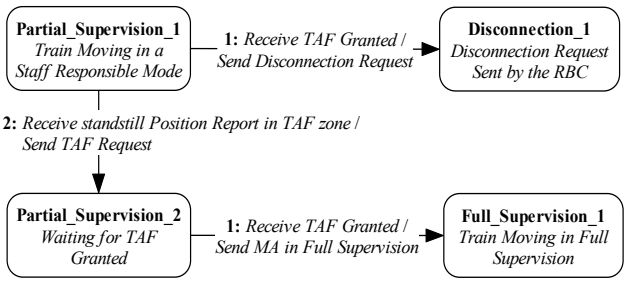


Figure. 7 The TAF scenario example.

The case (b) happens when the RBC receives the train-data message from the on-board in non nominal time instants or in scenarios different from the start of mission procedure, as requested by Unisig. The train data message contains train length, braking mass, shape limits, etc. which must be track compatible. Usually, the train changes its data only after an end of mission procedure, i.e after

[1] "Unisig compliant" means that system implementation respects the requirements contained in [1].

having disconnected from the RBC. However, as train data can be changed by the train driver in any moment at standstill, the RBC must be able to correctly manage such condition. The correct behavior designed for the RBC is the immediate verification of any train data received from the train against the maximum allowed boundaries. The specified functional test-cases require to verify that the RBC reacts with a disconnection request whenever it receives incompatible train data. Obviously, with a real on-board system which can change train data only after an End of Mission (in the correct implementation) it is not possible to execute such test. Therefore, the only solution to cope with such issues is to use the Anomaly Manager, which allows to simulate the behavior of any Unisig compliant on-board. This is a general need, as the Unisig specification often leaves freedom about system implementation. This implies that with a particular implementation of a subsystem it is not possible to stimulate other subsystems with all the possible conditions at the interface between them.

Finally, the (c) condition corresponds to an unexpected (or out of sequence) Track Ahead Free (TAF) message. The so called TAF procedure is mandatory in all cases in which the on-board has to pass from a partial supervision (e.g. due to route degradation) to a full supervision operating mode. With the TAF procedure the train driver, pressing a button on its MMI, notifies to the RBC the freedom of the track between the front-end of the train and the end of the track circuit occupied by the train, which can not be ensured by the RBC. In a correct TAF procedure, a TAF Request message is sent by the RBC to the train whenever it is able to assign it a Full Supervision movement authority; if the train driver acknowledges the TAF Request with the pressure of the TAF button, then a TAF Granted message is sent by the on-board to the RBC. In a non nominal case, the RBC could receive such a message without having previously sent a TAF Request message. This is risky, because in no way the RBC must send a Movement Authority to the train without the correct actuation of the TAF procedure: for instance, the train could not be in the so called TAF zone, the only one in which the TAF is allowed because of the limited human sight extension, or simply the on-board is acting in a wrong way. The test-cases specified for such condition have the aim to verify that the RBC state machine evolves correctly and protects from dangerous transitions: for instance, a possible design error could be to trigger the MA sending by the RBC in correspondence of the reception of a TAF Granted, without controlling that a previous TAF Request has been sent. The nominal as well as the abnormal test conditions have been represented in Figure 8, using the graphical formalism of the test specification methodology (see Section 1.2). A nominal on-board would be unable to reproduce the unauthorized TAF condition. Again, the use of the Anomaly Manager is the only way to easily overcome this problem. The configuration file of the Anomaly Manager can be prepared by making the simulator send a TAF Granted message before it reaches the TAF zone at the end of the track circuit: in fact in such a condition it is sure that the RBC does not output any TAF Request, whose sending is triggered by the reception of a position report message reporting train standstill in the TAF zone.

7. Conclusion and Future Work

ERTMS/ETCS is a complex railway control system featuring hundreds of functional requirements, many of them being safety-critical. To ensure a thorough functional verification of the trackside system, both a powerful test specification methodology and a flexible simulation environment are needed. The simulation environment must be able to execute all the specified tests, also regarding degraded track conditions and non nominal behaviors of the interacting entities. To achieve this aim, the nominal behaving simulators must be integrated with modules able to manage all the abnormal conditions, namely the anomalies. The complex and distributed simulation environment described in this paper, integrated with the so called Anomaly Manager, allowed the testing team to execute more than 1000 tests in a few weeks, using a proprietary scripting language in a completely automated environment. The test execution is still in progress but so far we did not encounter any test-case that we were unable to reproduce and automate in our environment. Moreover, regression tests are completely automated as the simulations can be repeated simply by using the same script files of the first test run.

We think that the simulation approach described in this paper could be easily generalized to deal with the simulation of communication anomalies for any system based on an open communication network. The general scheme adopted for the simulation of the anomalies, in fact, is needed and can be specialized for different systems whose robustness against message corruption, deletion, manipulation, etc. must be ensured, as requested by safety standards.

The simulation environment described in this paper is now being used to execute software stress tests in order to evaluate system performance in the worst operating case (the maximum number of on-board and IXL modules allowed for each RBC in the most stressful conditions). This is possible because the simulation environment as well as the Anomaly Manager are able to support any number of simulated trains.

References

[1] UNISIG ERTMS/ETCS – Class 1 Issue 2.2.2 Subset 026-1

[2] CENELEC: EN 50126 Railways Applications – The specification and demonstration of Reliability, Maintainability and Safety (RAMS), 1999.
[3] G. De Nicola, P. di Tommaso, R. Esposito, F. Flammini, A. Orazzo: A Hybrid Testing Methodology for Railway Control Systems. In Lecture Notes in Computer Science (LNCS) Vol. 3219 (ed. Springer-Verlag Heidelberg): Computer Safety, Reliability, and Security: 23rd International Conference, SAFECOMP 2004, Potsdam, Germany, September 21-24, 2004: pp. 116-135.
[4] G. De Nicola, P. di Tommaso, R. Esposito, F. Flammini, P. Marmo, A. Orazzo: A Grey Box Approach to the Functional Testing of Complex Automatic Train Protection Systems. In Lecture Notes in Computer Science (LNCS) Vol. 3463 (ed. Springer-Verlag Heidelberg): The Fifth European Dependable Computing Conference, EDCC-5, Budapest, Hungary, April 20-22, 2005: pp. 305-317.
[5] P. di Tommaso, R. Esposito, P. Marmo, A. Orazzo: Hazard Analysis of Complex Distributed Railway Systems. In Proceedings of 22nd International Symposium on Reliable Distributed Systems, SRDS2003, Florence, October 6-8, 2003: pp. 283-292.
[6] W. S. Heath: Real-Time Software Techniques. Van Nostrand Reinhold, New York (1991)
[7] K. Grimm: Systematic Testing of Software-Based Systems. In Proceedings of the 2nd Annual ENCRESS Conference, Paris (1996)
[8] I. Sommerville: Software Engineering, 6th Edition. Addison Wesley (2000)
[9] E. Dustin, J. Rashka, J. Paul: Automated Software Testing, Addison Wesley (1999)
[10] R. Esposito, A. Sanseviero, A. Lazzaro, P. Marmo: Formal Verification of ERTMS Euroradio Safety Critical Protocol. In Proceedings of FORMS 2003: Symposium on Formal Methods for Railway Operation and Control Systems, May 15-16, 2003, Budapest, Hungary.
[11] CENELEC: EN 50159-2 Railway Applications - Communication, signalling and processing systems - Part 2: Safety-related communication in open transmission systems, 2001.

Pasquale di Tommaso got his degree in Electronic Engineering at the Second University of Naples. Since February 2002, he worked in ASF on the verification and validation of SCMT on-board and ERTMS/ETCS trackside projects.

Francesco Flammini got his degree in Computer Science Engineering at the University “Federico II” of Naples, where he is currently a PhD candidate. Since October 2003, he worked in ASF on the verification and validation of SCMT on-board and ERTMS/ETCS trackside projects.