WHITE PAPER

Proceedings of 7th Transport Research Arena TRA 2018,
April 16-19, 2018, Vienna, Austria

AUTOMATIZED TESTING HIL SYSTEM FOR AGILE PRODUCT-DESIGN ENVIRONMENT

Michał Krześlak[a], Piotr Łuczak[a], Jerzy Kocerka[a]
aTritem Microsystems GmbH, Willy-Brandt-Strasse 5,
D-92245 Kümmersbruck, Germany

Abstract

The stricter safety requirements in the rail industry – implied by the EN 50128 standard – contribute to an increasing demand for testing rail vehicles and their subsystems, especially vehicle main controllers. Hardware-in-the-Loop method, which is commonly used in automotive and aerospace domains, has proved to be functionally useful. However, such systems would usually exceed the budget of a development project in the railway industry, due to short series manufacturing and multiple vehicle variants. To deal with this problem Tritem Microsystems has designed Virtual-HIL which decreases the overall cost and increases portability of this kind of a system. In this paper, we present both the classic approach and our groundbreaking system, along with a use case from one of our recent projects together with an automated testing framework built on the top of ourVirtual HIL.

Keywords: Hardware-in-the-Loop, software testing, simulation, Hybrid-HIL, Virtual-HIL, railway, functional safety

TRITEM MICROSYSTEMS GMBH
INFO@TRITEM.DE
+49 700 00 874 836
WWW.TRITEM.DE

Nomenclature

HIL Hardware in the loop
ECU Electronic Control Unit
ELMo Electro-mechanical Logic Modelling
UUT Unit Under Test
MMI Man-Machine Interface

1. Introduction

The increasing demand for testing and quality assurance in the railway industry, implied by the recent introduction of modern development standards, causes the railway rolling stock designers and manufacturers to seek cost- and time-optimal methods of software verification and validation. From the historical perspective, one of the milestones in this regard was the introduction of the EN 50128 standard. After the standard was published, key railway industry players quickly realized that the verification and validation methods, which had been in common use over the past several decades, (like careful manual testing of preproduction rolling stock specimens directly on the test tracks), would not meet the requirements of either the required testing effectivity level or the project time-frame. Those methods turned out to be simply too time-consuming and too expensive to allow those companies to successfully compete on the railway market which is undergoing a transformation. Taking into account several important factors like: increasing amount of vehicle functions implemented in software rather than hardware, high software reliability requirements, especially on higher software integrity levels (SIL2, SIL4), and increasing costs of performing tests on a real test track – both because of the cost of test track itself and because of the risk of damage – the leading rolling stock manufacturers have begun turning to techniques previously developed within the automotive and/or aerospace industry. One of the testing methods which seemed particularly appealing was the HIL (HardwareIn-the-Loop), where the physical controller is connected to a simulated plant in the laboratory in the identical way as it is connected to the real plant, allowing the tester to exercise functionalities of the control software even before it is applied in an actual vehicle, thus allowing to find errors quicker and with less effort (Alles et al. (1992) and (1994)). Although first attempts of adopting a classic HIL solution to a railway vehicle control software development process seemed fairly successful, some industry-specific considerations started to play an important role in the cost and flexibility of the development process. Unlike the automotive industry, where vehicles are manufactured in large, repeatable series, the railway vehicles are usually built in smaller batches. This difference is reflected in the overall project budget, and in turn it makes the cost of the HIL solution, and its simulation system in particular, a significant part of the overall project's cost. Moreover, such simulation systems turned out to be not quite as flexible as expected, and also relatively expensive to maintain. As a result, a demand for a new type of a HIL system, which would fit the specific needs of the rolling stock vehicle development process, has risen among

TRITEM MICROSYSTEMS
TRITEM MICROSYSTEMS GMBH
INFO@TRITEM.DE
+49 700 00 874 836
WWW.TRITEM.DE

the railway manufacturers.

In this paper, we would like to demonstrate how Tritem Microsystems tackled the problem of finding an optimal HIL solution by developing and deploying a Virtual-HIL system – founded on its unique simulation engine called ELMo – and how it overcame the obstacles related to this novel approach, what benefits such a solution brings to the overall cost and time of the vehicle development process, and how it be applied in other areas and industries. We will also discuss the existing HIL systems, their advantages and disadvantages, the complete test automation framework built around the Virtual-HIL, along with a case study demonstrating a real-world application of such a solution.

2. HIL systems

As stated in the introduction, the basic concept of a HIL system is to have the physical controller (also called ECU – Electronic Control Unit) connected to a simulated plant in the same way as it is connected to a real plant, i.e. such a virtual plant reacts with the ECU by simulating sensors, motors, actuators or other devices and – by communicating via various buses – assures that from the ECU's point of view the ECU is still in the real vehicle. A simulated plant can be divided into three major components: a test interface, a "machine," where the simulated plant is running, and I/O interfaces. The simulated plant is often run by real-time processor, which ensures deterministic times of simulation execution necessary for proper I/O operation. The I/O interfaces reflect the ECU communication with the real environment (via buses with for instance other ECUs and mechanical, electrical or pneumatic components). The operator interface provides the possibility to interact (send test commands, receive data and reports) with the real-time processor. This can be further used for performing both manual and automatic tests. Configuration management is also often done within this component. Fig. 1 depicts this concept.

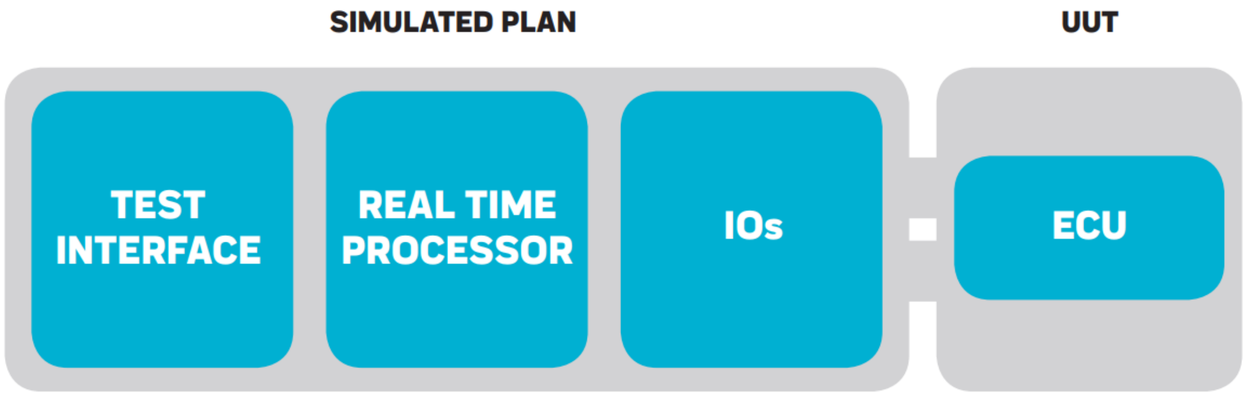

Fig. 1 HIL system with three major components and ECU hardware as Unit Under Test

As mentioned before, models and applications which run on the simulated plant reflect the tested ECU's environment. Considering the ECU as the main locomotive controller, the running models simulate the remaining part of the vehicle, which means almost the entire vehicle, including: driving controller, braking controller, door controller, electric (with circuit breakers, I/O modules, measurement devices etc.) and pneumatic schema. For users' (both testers and software developers) convenience, the operator interface needs to include the simulation of the real driver desk and driver displays (Man-Machine Interfaces – MMI) as well.

The common aim of HIL systems is to provide possibility to test ECU's software in the laboratory conditions. Over the years, the application of HIL approach has showed a multitude of benefits, out of which the following two major ones are particularly worth emphasizing:

• **Fault injection** – functional safety related functionalities can be tested in the laboratory before the ECU is used in the real environment. This reduces the risk of damaging any electrical or mechanical components that are controlled by the ECU under test, and increases the safety of people who would otherwise have to conduct the tests with real devices and vehicles. One example of such a scenario is testing of a functionality which triggers a protective action (e.g. driving interlock and information for the driver) when the temperature of one of the traction converters rises above a specified value. Raising the converter's temperature physically on the real vehicle could be really challenging or sometimes even dangerous. When using a HIL system, however, the tester can artificially set any value of the temperature readout, thus being able to test this functionality of the ECU in a thorough way without jeopardizing human personnel or expensive hardware.

• **Independent development** – provides the possibility to test algorithms and functions implemented in the ECU at an early stage of the vehicle and software development process. Thanks to this, the team developing the ECU does not need to wait for the delivery of any mechanical and electrical components or other devices that are required to build the prototype of the vehicle. All those functionalities can be developed parallelly by different teams and then integrated on the further stages, after being already initially tested in a common, simulated environment.

To leverage those benefits, the HIL system designed for the railway vehicle development domain has to have as many as possible quality features of such a system. To examine the quality of a HIL system, the following features and characteristics need to be analysed and taken into consideration:

• **Usability** – defines the ease with which the user interface of the virtual plant can be used. Whether it is user-friendly for testers and how fast they can learn to work with it.

• **Efficiency** – the simulated part should ensure that its loop is running at the high sampling rate. Sampling rates (loop cycling times) may differ due to various needs and applications of the HIL system; however, in general, the measure of the simulation accuracy can be defined by the possibility of high sampling rate.

• **Maintainability** – projects, especially in railway industry, last for long periods of time; thus, it is important that both hardware and simulated parts of the HIL are easy to maintain and upgrade, if necessary.

• **Scalability** – related to efficiency and maintainability; this measure defines if the system architecture can easily bare an increase of load (such as adding new functions, models, data etc.).

• **Portability** – in terms of software testing portability defines the ease with which a tested application can be moved and used in different environments (e.g. different Windows systems). In the case of HIL, we consider it as the ease to reuse an existing system (mainly, a virtual plant) for new projects (e.g. different configuration of the same vehicle or even a new vehicle). Basing on our experience, it is one of the most important characteristics, and several sub-characteristics could be identified here.

This is why we focus on analysing existing HIL systems and describing our Virtual-HIL system with our innovative ELMo engine below.

In the following chapters, we will analyse the already available HIL approaches and show how the VirtualHIL synthesizes what is best about all the other approaches to eventually, thanks to its unique features, become - from the vehicle vendors' point of view - an investment rather than a cost generating element of a given project.

2.1. Classic HIL

The complexity and configuration of a HIL system differs from one project to another. It usually depends on the particular domain (e.g. automotive, aviation or, discussed here, railway); however, the configuration depends on the purposes of the HIL system, which is mainly related to the testing scope. In a basic concept, there is one ECU which is the unit under test and all the other elements or components of the system belong to the virtual plant (this includes other

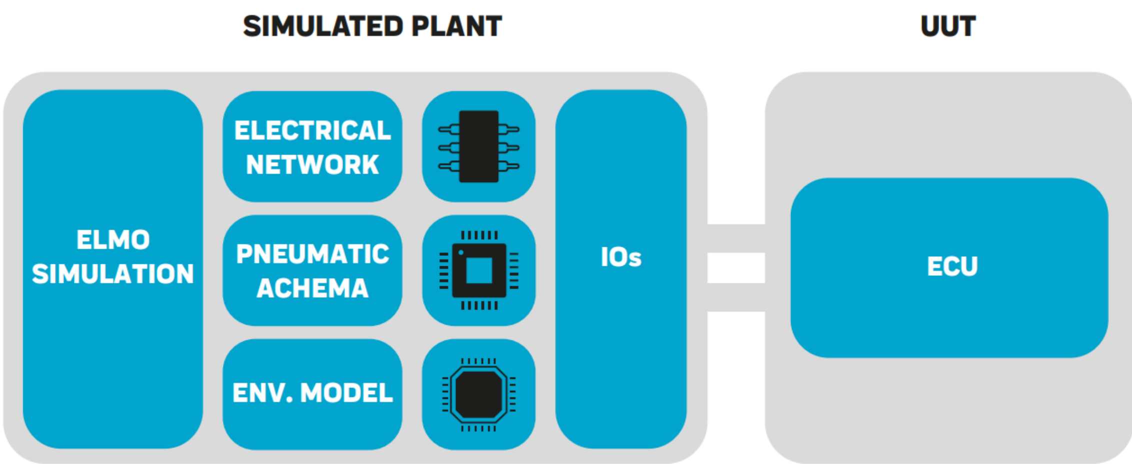

Fig. 2: HIL system with three major components and ECU hardware as Unit Under Test

ECUs, electrical schema elements, etc.) (Fig. 2). Some norms (like the aforementioned EN 50128) and test processes require a different scope which affects the HIL configuration, e.g. the necessity of performing integration tests between two ECUs (for instance: the main train controller and braking controller). Such a HIL system may then contain two or more hardware ECUs, simulated plant and sometimes also an MMI, which can be quite often treated as an inclusive part of the main controller.

A different scope of the planned tests and an aim of the HIL entail various HIL configurations in terms of what exists as a hardware component, and what is simulated. If one needs to test, for instance, how the braking controller and the pneumatic components work together, then those become the hardware parts of the HIL, and all the rest of the elements that are necessary for the system to run (like some functionalities of the main controller, driving controller or electrical schema elements) become part of the simulated plant (Fig. 3).

Not everything needs to be simulated, only those parts which are required to ensure proper virtual environment for the hardware part.

Dhaliwal et al. (2009) define a similar division for different HIL configurations. They have divided it into three levels: signal level interface (the hardware part is only the electric motor controller), electrical level interface (the hardware part additionally includes power stages), mechanical level interface (additional load motors, dynamometers, etc.). Which is the equivalent to the generalization presented in Fig. 4.

As mentioned before, the complexity of an HIL system also depends on the working domain. Bocker et al. (2012) mentioned that the difficulty in developing HIL for railway vehicles results from the fact that such vehicles contain more than one traction motor and power converter, unlike in the automotive industry (electric or hybrid vehicles). Taking into account that the railway vehicles operate often as multi-consists (i.e. a

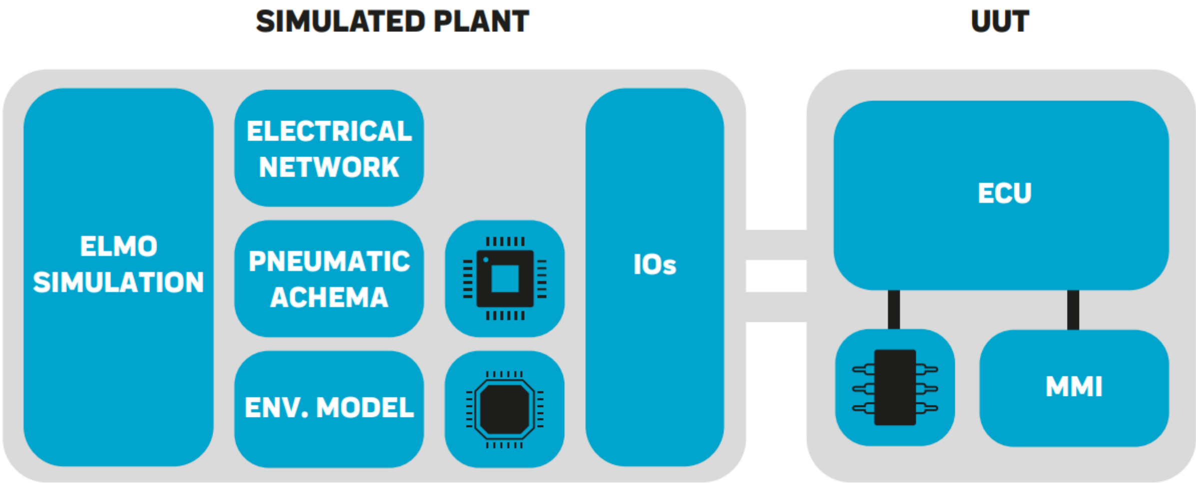

Fig. 3: HIL system with additional ECU and MMI as Unit Under Test

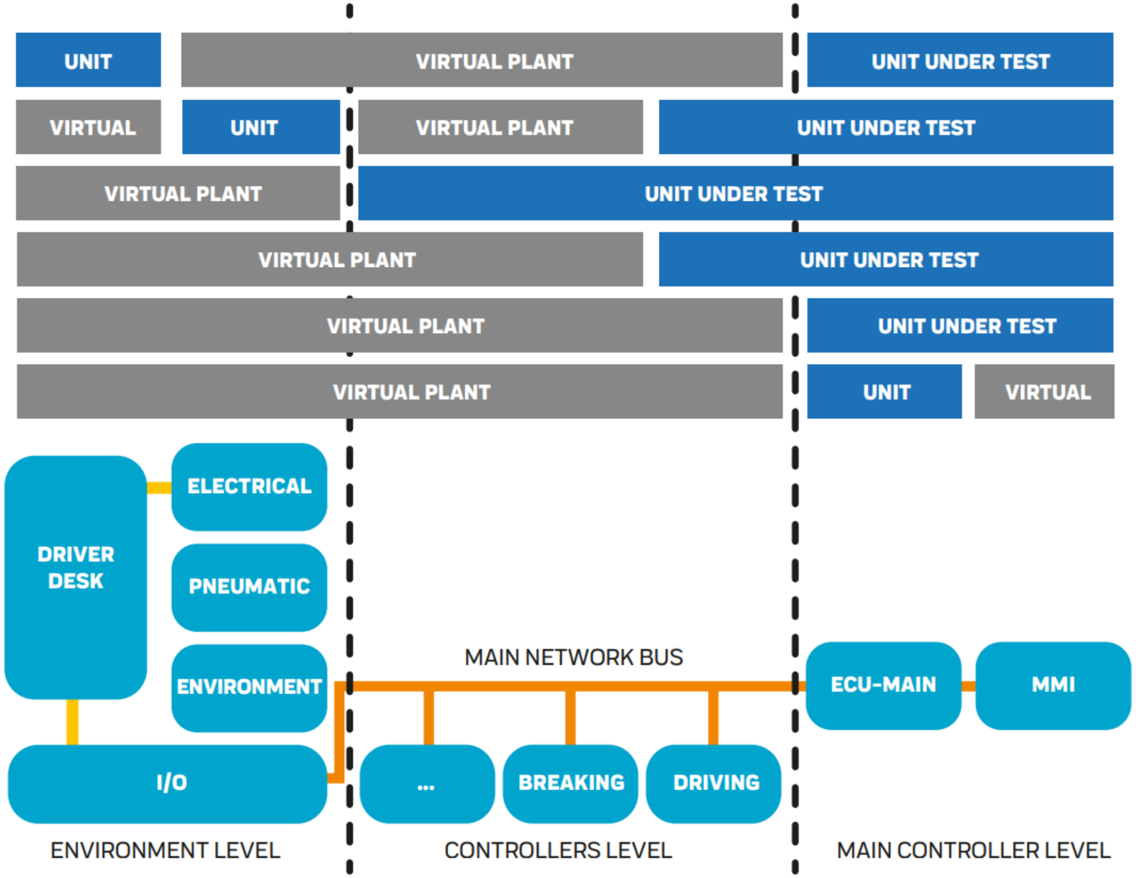

Fig. 4: Different HIL configurations with three levels: environment, controllers and main controller level

couple of trains connected) makes the development of a proper, efficient HIL for the railway industry even more complex. Within the same paper, the authors discuss another limitation or difficulty which needs to be tackled in the railway domain, i.e. the diversity of the product line, so that it offers various train drive concepts, multitude of model variants and other vehicle parameters, which is exactly what has already been highlighted in the introduction part.

2.2. Hybrid HIL

The same concept of hybrid HILs seems to be differently defined and understood in diverse fields and papers. Classic HIL approach (described earlier as ECU under test and virtual, simulated plant) may be in some cases already treated as a hybrid HIL (Qi et al. (2017)); comparing this approach with a Software-in-the-loop, aside from the software, here is a piece of hardware with a tested application running in. In articles wrote by Dhaliwal et al. (2009) and Bocker et al. (2012) a classical HIL combined with an FPGA based simulator is described as a "hybrid HIL." Since the efficiency of the HIL is at the same time one of the most important features and the biggest challenge, proper computation load balancing between the HIL and FPGA part becomes crucial for this kind of systems. However, the usage of FPGA introduces longer synthesizing ti-

TRITEM MICROSYSTEMS
TRITEM MICROSYSTEMS GMBH
INFO@TRITEM.DE
+49 700 00 874 836
WWW.TRITEM.DE

me and makes it more complex to recompile the model using different parameter sets for different configurations, which – in turn – impairs portability (Bocker et al. (2012)). Another definition of a hybrid HIL is based on the fact that more than one hardware element is used. As mentioned earlier, the aim could be, for instance, to perform integration tests between two ECUs or ECU with mechanical, electrical or pneumatic elements of the vehicle (Fig.2). Sometimes, it is easier for HIL vendors to leave real hardware, instead of simulating it and including in the virtual plant. The latter is less loaded (so the efficiency is less compromised) and there is less effort for virtual plant development; however, such a set may be limited in terms of fault injection possibilities, considering the fact that some inputs to the tested ECU are directly connected to this additional hardware. Moreover, this additional hardware also should be ready and tested (to make sure that it does not introduce false failures), i.e. to fully assemble such multi-hardware HIL, all of the elements must be finished, which limits the major benefit of this approach, namely the possibility to develop and test the ECU independently from other vehicle elements. The most important benefit, however, is that the simulated device usually is much cheaper than the real one. For instance, in case of hybrid electric vehicles, the most expensive component is the battery (Dhaliwal et al. (2009)); thus, this is usually the part of the hybrid car that is simulated. Moreover, the authors also indicate that the software running on ECU has to be robust and reliable before it is connected to the batteries. When a HIL system is in place, the development phase of the ECU software can be performed in the laboratory, without using high voltage or current

Basing on our experience, the definition of hybrid HIL as a simulated plant with FPGA seems to be the most suitable one. The HIL system designed and developed at Tritem also supports such a configuration, and also just like Dhaliwal et al. (2009) and Bocker et al. (2012) we have found it to be a very useful method to increase the sampling rate of a virtual plant. However, it remains beyond the major scope of interest examined in this paper, so we will focus here more on the classic HIL approach, and different degree of its characteristics depending on the configuration and complexity.

2.3. Virtual-HIL

To fulfil all the specific requirements of the railway domain and provide sufficient agility, Tritem had to develop yet another kind of HIL system, taking what is best about already existing approaches, and combining it with its own solutions and expertise.
The HIL system designed at Tritem is a system that is primarily highly portable. It allows to change the existing configuration quickly to provide a different testing scope and level. Therefore, the Virtual-HIL permits to morph, for instance, a system with the main vehicle controller and an HMI into the system with an additional driving controller. This offers a possibility to perform integration tests between those two controllers (like from HIL presented in Fig. 2 into the HIL

from Fig. 3) at any time. The high degree of portability in Virtual-HIL allows not only to extend the current HIL architecture by adding new hardware units quickly, but also to re-use the existing modules and models for completely new vehicles. Over the past years, Tritem has successfully deployed several such HIL systems for different kinds of vehicles. One of the examples that has proved the usefulness of modularity was when an existing HIL built for an electrical locomotive was used as a basis for an HIL for a locomotive with a diesel engine. In this scenario, more than 80% of the components of the original simulation could be reused without modification. The advanced portability and modularity of the Virtual HIL was achieved thanks to the key part of the Virtual- HIL, which is: ELMo. An ELMo engine makes it possible to easily synthesize various existing models together, creating a new type of vehicle. Another advantage of ELMo engine is the possibility to reuse and parse some of the pre-existing project data, like electrical schema or interface signal definitions. The significant example, used in almost all Virtual-HIL applications, is parsing and synthesizing the electrical schema (wiring diagram) of the vehicle under test into a virtual plant. This feature makes ELMo extraordinarily generic and configurable; what is more, a lot of time and effort is saved while setting up a new HIL for a new vehicle.

Moreover, at an early stage of the project, when the electrical schema tends to change frequently, all the developers had a working simulation at their disposal within minutes after a newer revision of the schema was published. This allowed to keep the software development and testing loop smooth and continuous. High degree of portability also allows to easily extend the existing HIL system by replacing simulated elements with their hardware counterparts (Fig. 4 presents different possible configurations with hardware and/or virtual elements). ELMo gives also a high degree of scalability to the HIL system, i.e. the reverse reconfiguration. HILs created with a lot hardware can be reduced to the minimum amount of hardware units replacing those elements with proper models (synthesized by ELMo) without losing the efficiency of the simulated plant. As mentioned earlier, in some areas the HIL configured with more than one hardware unit can be defined as a Hybrid HIL. In this sense, the Virtual-HIL's hybridity does not manifest itself with the fact that it is possible to set it up with different hardware elements or controllers, but with the luxury to reconfigure the existing system with virtually no effort and time.

2.3.1 Virtual HIL Application

Tritem Microsystems together with one of its clients, a major player on the railway vehicle development market, have built a Virtual HIL test system for a train control and management system (TCMS). The main goal was to build a system to test the main vehicle control unit (VCU) The system had to be modular (i.e. could be used for various subsystem tests) and flexible (i.e. could be used for different projects or even vehicle kinds in the future).

The system was based on Tritem ELMo and National Instruments VeriStand software running on an industrial PC (or PXI) with a real-time operating system (NI ETS formerly known as Phar Lap ETS). The communication to the unit under test (UUT) was realized with ProfiNET network. The modularity of the system allowed to connect various additional subsystems for integration testing (e.g. a brake control unit). In such a case, the system was simply extended with additional I/O modules, communication buses (e.g. CAN) and FPGA boards, if the there was a necessity for a high speed signal generation or signal conditioning. The I/O modules and FPGA were connected via EtherCAT (deterministic Ethernet) allowing developers and testers to change the setup very quickly (just by plugging in the Ethernet cable).

Due to the fact that the system was based on software (ELMo models, VeriStand models, FPGA code), it was a matter of minutes, if not seconds, to reconfigure it into a completely new setup (for example with different modules or for a different project). To make the process even simpler Tritem, had developed a special tool chain for creating, maintaining and deploying various configurations. The whole system proves the idea of the Virtual Hardware-In-the-Loop systems, giving our client the ability to test and debug their software in the laboratory and perform integration testing even before the whole train is built. The project was a huge success for both Tritem and its customer, and has returned the initial costs to the customer multiple times, making the HIL system an investment bringing profit to the company rather than a regular project cost.

2.3.2 Virtual HIL and Automatic Tests System

To make the usage of Virtual-HIL even more convenient, it can be made part of a bigger Automated Testing System.

On top of the ELMo-based Virtual-HIL, Tritem builds a testing toolchain which is largely based on National Instruments technologies. It provides the user with the means of test script management, remote execution and results processing. When integrated with the customer's ALM/PLM infrastructure and tools – this could be IBM DOORs, Polarion or virtually any other system – it ensures full traceability between testing artifacts and the requirements (Fig. 5).

The test framework allows to develop test scripts using domain specific language, i.e. the one which is usable for domain experts who can create tests easily, without having advanced programming skills, and it is readable also to the business management, so also for the stakeholders who can be interested in what particular test is carried out and what the result is.

Assuming that the test scripts are written and stored in the system, the framework operates fully autonomously, meaning that to execute tests of a given functionality on a particular software/hardware configuration, it is enough to select the appropriate configuration in the test management system, and schedule the tests to run at a given time in a given location. The automation system will then arrange all the necessary configuration files, upload software, feed ELMo with the relevant electrical schemas and models, and eventually execute tests in that environment.

The practical efficiency of an ELMo-based test automation system can be illustrated with an example of one of Tritem's other customers' success stories: the complete pre-release regression tests of the software of the main vehicle system controller (one version only!) are performed within 9 days, i.e. 216 hours, in the 24/7 work mode. It is worth emphasizing that it is not necessary to involve any human personnel in this task in the aforementioned time period. The system executes more than 3000 test cases within this time.

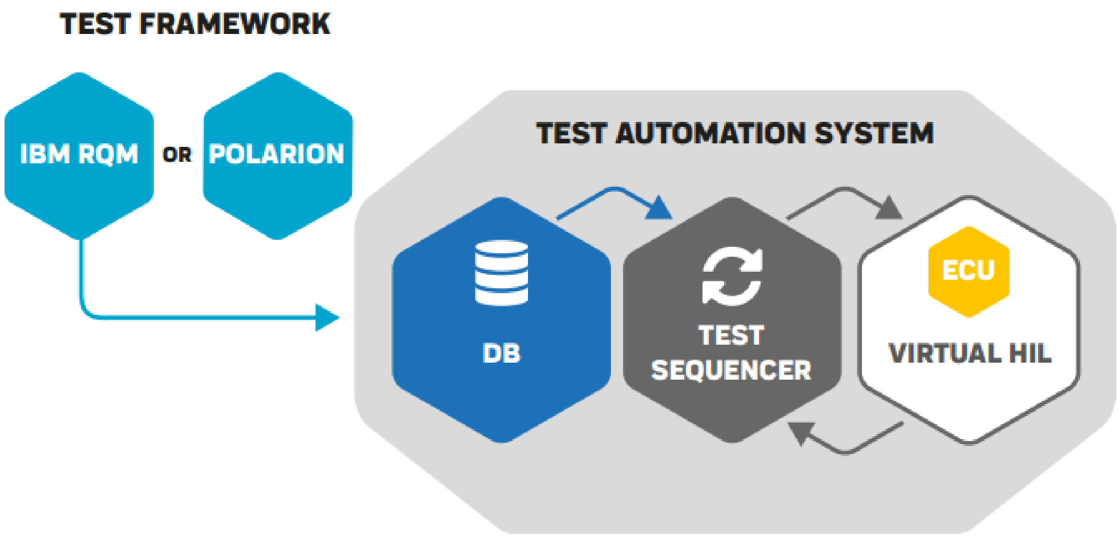

Fig. 5: Test Framework with ALM/PLM tools, Test Automation System and ELMO-based Virtual-HIL

3. Summary

The stricter safety requirements in the railway industry – based on the EN 50128 standard – contribute to the increase of the demands concerning testing of rail vehicles and their subsystems, especially their main controllers. Hardware-in-the-Loop method – commonly used in automotive and aerospace industries – has proved to be functionally useful. However, the leading rail vehicle manufacturers have realized that there are specific conditions associated with their field: rail vehicles are built in short series in comparison with the automotive industry, and thus require tailored solutions. It contributes to the fact that the cost of HIL-type vehicle simulators usually exceeds the budget of a typical development project in the rail industry.

The foundation and the key part of the Virtual-HIL automated testing system proposed by Tritem is a virtualplant engine called ELMo, which stands for Electro-mechanical Logic MOdeling. ELMo makes it possible to design and emulate a plant of a HIL system in real time, without the complicity, cost and overhead of conventional HIL simulators. It automatically creates the plant structure – i.e. a virtual vehicle is created by downloading the wiring diagrams of the vehicle under test. Furthermore, it allows for the coexistence of real hardware subsystems, emulated subsystems and components as well as simulation blocks. This feature makes ELMo extraordinarily versatile, and applicable not only in the field of system validation, but also within the agile design & development process. ELMo supports ad-hoc reconfiguration in order to create any other – sometimes totally different – vehicle variant. Its compactness, as well as low unit cost, makes it possible to use numerous systems of this kind for various purposes and projects by providing the component database with varied emulation models.

Furthermore, ELMo-based systems could work just as well in other domains, showing significant advantages during the development process of all kinds of vehicles and vessels which are typically built in short series and multiple variants.

4. References

Alles., S., Swick, C., Mahmud, S., 1992. *Real time hardware-in-the-loop vehicle simulation.* Instrumentation and Measurement Technology Conference, 1992. IMTC '92., 9th IEEE.

Alles., S., Swick, C., Hoffman, M., Mahmud, S.M., Feng Lin, 1994. *A real-time hardware-in-the-loop vehicle simulator for traction assist.* Int. J. of Vehicle Design, 15, 6.

Bocker, J., Sun, M., Cao, Z., 2012. *High Fidelity Hybrid Hardware-in-the-Loop Simulator with FPGA and Processor for AC Railway Traction.* PCIM Europe, Nuremberg, 110

Dhaliwal, A., Nagaraj, S., Santhosh, J., 2009. *Hardware-in-the-Loop Testing for Hybrid Vehicles.* Evaluation Engineering.

Qi, Ch., Zhao, X., Gao, F., Ren, A., Hu, Y. 2017. *A Robotic Hardware-in-the-Loop Simulation System for Flying Objects Contact in Space.* IAS 2016, Advances in Intelligent Systems and Computing, 531, pp. 725-735

Michał Krześlak
Testing Group Leader

A graduate of the Silesian University of Technology in Gliwice, Faculty of Automation and Robotics. He also holds a Ph.D. in biocybernetics.

At Automatix, the competence center for Tritem company, he manages a team of specialists dealing with software testing engineering as well as with R & D of methods and techniques connected with this field. As he emphasizes himself, the specialists in his team can be called triathlonists: they are programmers, first users and of course testers.

Piotr Łuczak
Software Team Leader

A graduate of the Silesian University of Technology in Gliwice with the specialization in Computer Control Systems.

At Automatix, the competence center for Tritem company, he manages the team responsible for software engineering, controller programming and integration of the company platform with systems used by clients.

Jerzy Kocerka
Simulation Group Leader

A graduate of the Warsaw University of Technology; passionately keen on NI LabVIEW system-design software and National Instruments technology.

At Automatix, the competence center for Tritem company, he manages a team of technicians who develop Hardware-In-the-Loop systems, simulations and software used to support testing processes.