Scenario-Based Validation & Verification, the ENABLE-S3 Approach

Joan J. Valls, Miguel García-Gordillo, Sergio Sáez

Instituto Tecnológico de Informática, Valencia, Spain; email: {jvalls, miguelgarcia, ssaez}@iti.es

Abstract

Automated systems can be found on many current vehicles, either land, air or maritime. The reliability, safety and robustness of these systems is extremely important, hence validation approaches need to adapt to the ever-evolving necessities of the industry. The ENABLE-S3 architecture addresses the problem of extensive testing by introducing a set of tools and methodologies that can be used to build up a testing environment for different domains. In this manuscript, a special focus is given to the solution developed for the Reconfigurable Video Processor from the aerospace domain.

Keywords: ENABLE-S3, validation, verification, scenario.

1 Introduction

Advances in the development of automated systems can be assessed with the millions of test kilometers that have already been travelled by automated vehicles on public roads. These kind of technologies are leading to improvements in safety, in a more environmental friendly and efficient driving, as well as reducing the number of accidents. Similar statements can also be said for other highly Automated Cyber Physical Systems (ACPS).

Demonstrating the reliability, safety, and robustness of this technology is an arduous task that requires a thorough analysis of the system behaviour under all conceivable situations and potential environmental conditions. There is a lack of cost-effective, commonly accepted verification & validation (V&V) methods. This has been identified as the main roadblock for product homologation, certification and later commercialisation. For instance, some studies [1] state that more than 100 million km of road driving are required to prove that an automated vehicle is statistically as safe as a manually driven one.

In the current digital age, products of the aerospace industry have to fulfill the needs of the user, hence it needs to keep up with the pace of technological developments. Even though aerospace products - specifically satellites and aircrafts - have a very long lifetime, they need to show higher flexibility and better performance in their usage. New solutions are expected to be cheaper, reliable and reach the market in less time. In the case of satellites, for example, this is due to the pressure of NewSpace [2]. The use of Commercial Off-The-Shelf (COTS) components is one of the most evaluated ways to achieve better performance and lower costs. However, their applicability needs to be tested and adapted, e.g. to the space environment, and their suitability in terms of safety and security needs to be checked, e.g. in aerial transport. In the ENABLE-S3 project a lot of effort has been put into addressing the problem in extensive testing these components to guarantee that a sufficient confidence in its reliability and security is achieved. New approaches in system validation are in demand.

The remainder of this manuscript is organised as follows: Section 2 introduces the architecture and methodologies developed during the ENABLE-S3 project. Section 3 presents the difficulties encountered in the aerospace domain and in the Reconfigurable Video Processor use case, particularly. Section 4 describes the scenario-based virtual validation & verification approached followed in the use case. Section 4.2 summarises the test framework to evaluate the physical system. Finally, Section 5 offers some concluding remarks.

2 ENABLE-S3 Approach

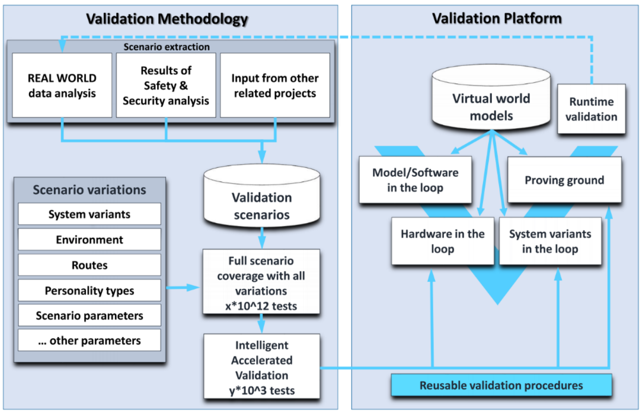


Figure 1: ENABLE-S3 validation toolchain architecture

The aim of the ENABLE-S3 project is to provide the required means for the verification & validation of ACPS. The solution pursued in the project is the identification of relevant scenarios, the automatic derivation of manageable sets of test cases from scenarios as well as the application of automated virtual V&V approaches in combination with physical testing. A consortium of 68 industry and research partners from different application domains (automotive, aerospace, rail, maritime, health and farming) have joined their forces to develop the required technology bricks. Due to this diversity of

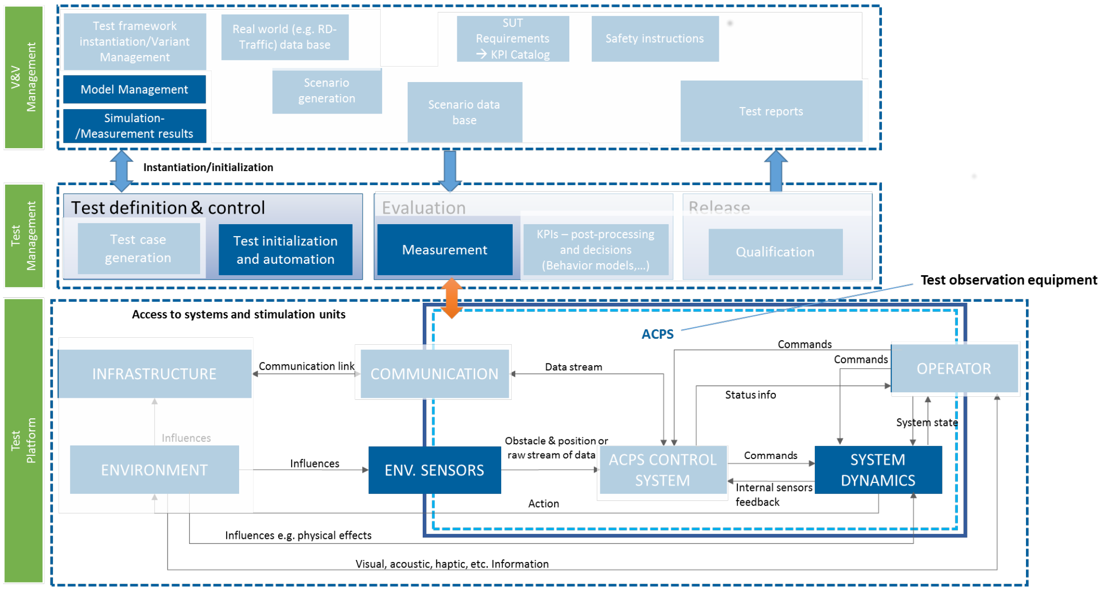

V&V Management
Test framework instantiation/Variant Management
Real world (e.g. RD-Traffic) data base
SUT Requirements → KPI Catalog
Safety instructions
Model Management
Scenario generation
Scenario data base
Test reports
Simulation-/Measurement results
Instantiation/initialization
Test Management
Test definition & control
Test case generation
Test initialization and automation
Evaluation
Measurement
KPIs – post-processing and decisions (Behavior models,...)
Release
Qualification
Test observation equipment
Test Platform
Access to systems and stimulation units
ACPS
Commands
Commands
INFRASTRUCTURE
Communication link
COMMUNICATION
Data stream
OPERATOR
Status info
System state
Influences
Obstacle & position or raw stream of data
ENVIRONMENT
Influences
ENV. SENSORS
ACPS CONTROL SYSTEM
Commands
SYSTEM DYNAMICS
Internal sensors feedback
Action
Influences e.g. physical effects
Visual, acoustic, haptic, etc. Information

Figure 2: ENABLE-S3 testing environment in the aerospace use case

partners and application domains, the project does not aim for a single common, generic software solution. The idea pursued was the development of a common methodology, a basic verification and validation toolchain architecture (Figure 1) and a set of reusable technology bricks, which can be used to build up a testing environment for use cases in different industry domains (Figure 2). The complete results of the project have been published [3]. The remainder of this manuscript summarises the work done in the aerospace domain use case. More detailed information on this use case can be found in [4].

3 Aerospace Domain Use Case

The main focus of the aerospace domain use case is the improvements in the validation and verification processes of a Reconfigurable Video Processor (RVP) that will be sent to space missions.

One of the main problems of autonomy in space applications is that once a mission is in orbit it is very difficult, even neigh impossible, to replace a processing module on board. Application-Specific Integrated Circuits (ASICs) and antifuse-based FPGAs are the most common solutions for the vast majority of space digital systems. Problems related to these technologies are the non-recurring engineering costs and the lack of flexibility that is demanded in the New Space which is currently being defined. For that reason, SRAM-based FPGAs that serve as programmable devices with shorter design cycles and reduced NRE could alleviate the aforementioned inconveniences. Additionally, modern FPGAs also offer the possibility to be reprogrammed on-the-fly which makes them more interesting for remote long-term space missions.

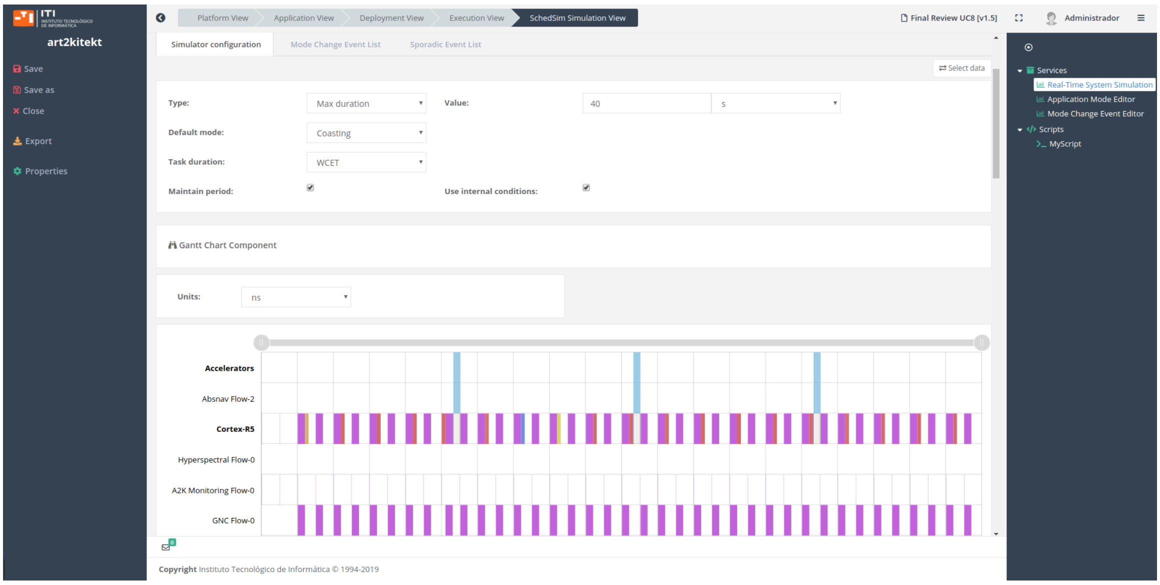
However, in space missions, there are some unique environmental challenges that need to be accounted for and that may have a large impact on the utilization of these technologies. Despite the high performance, flexibility and low design costs, the volatile nature of SRAM-based FPGAs makes them highly susceptible to radiation effects. When a particle hits and SRAM-based device, the content of one or several cells may change. When this event happens, the implemented functionality may also change which can have catastrophic consequences. Hence, commercial parts employing these FPGAs cannot provide a reliable hardware for space environment since they have not been designed following secure radiation-hardened and fault-tolerant design processes.

The ARTICo3 architecture [5] is a hardware-based processing architecture for high-performance embedded reconfigurable computing. It uses Dynamic and Partial Reconfiguration (DPR) in SRAM-based FPGAs as its technological foundation. The architecture supports and enables run-time adaptable implementations of data-parallel algorithms. Two different types of parallelism can be exploited using ARTICo3: (1) by using several replicas of the same hardware accelerator it provides data-level parallelism, and (2) by using different hardware accelerators it enables task-level parallelism. Although these features offer fault tolerance in the reconfigurable partition, additional mechanisms are required in a safety-critical context, as the space scenario of the use case.

To correct transient faults in memories, some techniques, generally called scrubbers, are utilised. In the RVP they have been implemented in different layers: real-time processors (ARM Cortex R5), platform management unit (PMU), and dedicated hardware cores inside the FPGA.

A wide range of Cyber-Physical Systems (CPS) applications can be developed with the ARTICo3 framework. The applications should support HW/SW partitioning, so that certain tasks can be offloaded to and executed in hardware accelerators. Two algorithms have been implemented in regards to the use case, which follow the requirements of the framework.

On-board image compression techniques are mandatory in space remote sensing missions to reduce data size prior to



Type: Max duration

Value: 40 s

Default mode: Coasting

Task duration: WCET

Maintain period: ☑

Use internal conditions: ☑

Units: ns

Copyright Instituto Tecnológico de Informática © 1994-2019

Figure 3: Art2kitekt Scheduling Simulation Tool

sending them to ground stations where they are processed. In this case, a lossy extension of the Consultative Committee for Space Data Systems 123.0-B-1 Lossless Multispectral and Hyperspectral Image Compression algorithm has been developed [6]. The algorithm provides a trade-off between its compression efficiency and the design complexity. The output data constitute a variable-length encoded bitstream from which the original image can be fully recovered. It uses a scheme based on prediction and entropy coding of the resultant prediction residuals, i.e. the differences between each input sample and its corresponding prediction value.

Guidance, Navigation and Control engines will allow autonomous navigation of spacecrafts, including traveling to planets or asteroids surfaces, orbiting around those stellar bodies, etc. Accurate positioning is required to enable pinpoint landing ability. Several vision-based navigation algorithms have been implemented, and since they are computationally intensive, the hardware acceleration that can be obtained thanks to the accelerators in ARTICo3 will be essential. Specifically the algorithms that have been developed are: Absolute Navigation Algorithm, Relative Navigation Algorithm, and Stereo Vision Algorithm.

4 Scenario-based virtual V&V

Since the RVP for space missions cannot be validated under real conditions, a scenario-based virtual validation campaign is performed. This virtual V&V campaign aligns to the following main objectives of ENABLE-S3:

- reduction of expensive testing time in nuclear facilities;
- optimization of the test setup through lab testing and prior to real radiation testing;
- strengthen critical parts of the design in early stages of the development;
- optimization of the component qualification effort.

4.1 Model-in-the-Loop

First step of the validation will be done at the model level. The RVP is modelled and analysed in order to discard unfeasible scenarios early on. A test scenario is planned, inspired by the different stages a spacecraft may encounter during a real space mission. The model includes the adaption capabilities of the RVP, i.e. the recovery mechanism/fault mitigation systems and the reconfiguration mechanisms to handle the different operational modes and its transitions.

4.1.1 Platform and application models

The art2kitekt tool suite is used to assist in the V&V process, more concretely, its modelling features that help in the design process of high integrity systems with real-time constraints.

A collection of components such as processors, memories, buses and devices are offered by the framework and allows the engineer to build an heterogeneous system that suits their needs. The desired system model is built by creating instances of defined component types (e.g. the user can define their own processor types) and by connecting them through buses. The current version of the framework allows the instantiation of processing devices, i.e. components that are not processors but which are able to execute code (e.g. an accelerator). This also includes the programmable devices part of the ARTICo3 architecture.

The specification model that describes the application consists of a series of independent execution flows comprised of several activities. In some contexts they are referred to as end-to-end flows and tasks. These flows have different activation patterns, i.e. either a periodic activation or an sporadic activation after certain event has been issued. Each flow has several activities that model the inner functionality and the precedence relationships between them. All activities can be refined from coarse to fine-grain modelling as the engineer considers necessary.

4.1.2 *Scheduling Simulation Tool*

Schedulability analysis is one of the most important evaluations of a system, especially in hard real-time systems [7, 8], in which missing a deadline can lead to catastrophic consequences. There are several formal techniques to analyse the worst-case response time of tasks [9]. In some scenarios, these techniques do not always provide exact solutions, for example in distributed hard real-time systems [10]. They work with assumptions such as all tasks being independent or requiring some restrictions in order to apply them, hence results are pessimistic. This, in turn, brings the inefficient use of the computing power of real-time systems in order to guarantee the feasibility of its schedulability.

A complementary approach to performing off-line analysis with formal techniques is the evaluation of the real-time system through simulation of its real-time behaviour. Although simulation cannot assure the validation of a system, it can help with the study and understanding of its behaviour and limits. Some of the advantages offered by simulation that make for an interesting tool are:

- there is no probe effect (no disturbance) on the system due to instrumentation, which can be problematic in real-time systems;
- the engineer can replay scenarios with ease in order to evaluate how changing the studied tasks, or even the executing platform, may affect the results;
- the effect of non-static or non predictable events can be studied.

A new scheduling simulation tool has been integrated into the art2kitekt framework, a screenshot of which is depicted in Figure 3. The implemented tool adapts to the specific requirement of the use case. For instance, it has the capability of simulating non periodic events such as the presence of a fault in the system due to radiation. Also, the tool incorporates functional modes and its mode changes, in order to represent and analyse the different behaviour of the system during each of its mission phases, since it is essential to guarantee the fulfilment of the deadlines not only during each operational mode, but also during the transitions between them. In this manner, the engineer can model several scenarios to have a more detailed analysis and this leads to a reduction of the test procedure by detecting unfeasible scenarios in early stages of the development.

4.2 Hardware-in-the-Loop

In the next steps, the aforementioned simulation models must be implemented using functional software for testing in the final hardware. The model components are step by step replaced by software components (SiL, Software-in-the-Loop) and executed in the real hardware components (HiL, Hardware-in-the-Loop), and finally, the overall system is tested. This methodology requires a flexible and safety simulation framework with the capability of communicating with the hardware platform, complying with the real-time requirements of the System Under Test (SUT). This framework should help in integrating the different technology bricks involved in the tests and in collecting the results provided by them.

4.2.1 *Test Framework*

The ENABLE-S3 architecture defines the generic test framework that is comprised of the test execution platform and the test management. Different applications are involved in these processes and are coordinated by art2kitekt, in order to generate the test scenarios and to execute the SUT with the required test configurations.

Figure 4 depicts the validation platform in the aerospace use case that has been developed over a board based on a Xilinx Zynq Ultrascale+ MPSoC. It is composed of (1) a Real-Time Processing Unit (RPU), (2) a Programmable Logic FPGA-based unit (PL), and (3) a general Application Processing Unit (APU). The RPU and the PL are in charge of executing the SUT, and the APU is responsible of executing the embedded part of the Test System (TS). Both, the SUT and the TS, coexist in the same system but running in different domains.

A bidirectional communication protocol between the TS and the SUT has been implemented thanks to the capabilities of the MPSoC: the shared memory and the Inter Processor Interrupt device. With the aim of sending large messages between processors using a zero-copy approach and a low-latency transmission. This communication permits (1) to configure the SUT, (2) to coordinate the test execution with the Test System, and (3) to collect the output measures to allow later processing and visualization.

The test scenario is composed of several Application Processes (APs) running in the TS processor, that are connected with their counterpart in the SUT processor. The different APs involved in the tests are the gateway between the SUT and the external applications, in charge of simulating the sensors and the data providers necessary to generate the scenarios. Thanks to this architecture, the TS provides an interface to help in the integration of independent test applications.

Furthermore, the system coordinates the different APs and the SUT, using a common command interface, that permits to configure the whole system with the same parameters and to coordinate its execution. The TS coordinator, based on a state machine approach, allows the application to be commanded, waiting for them and avoiding APs from breaking the test timing.

In the aforementioned test platform, art2kitekt offers both, the low level architecture to build the test system, and the user interface to configure the scenarios and to visualise the test results. This tool suite has the goal of helping the engineer to control the system in the whole V&V process.

4.2.2 *Monitoring Tool*

Observing the behaviour of a real-time system helps the engineer in several ways in the V&V process. In early phases of the development, observed execution time, mixed with the estimated one, can be used to analyse the system and to ensure the fulfilment of the temporal constraints. In addition, comparing the observation with the expected behaviour helps to find incorrect implementations and to improve the initial model.

The art2kitekt toolsuite has been extended to provide a run-time monitoring tool [11] with the capability of collecting

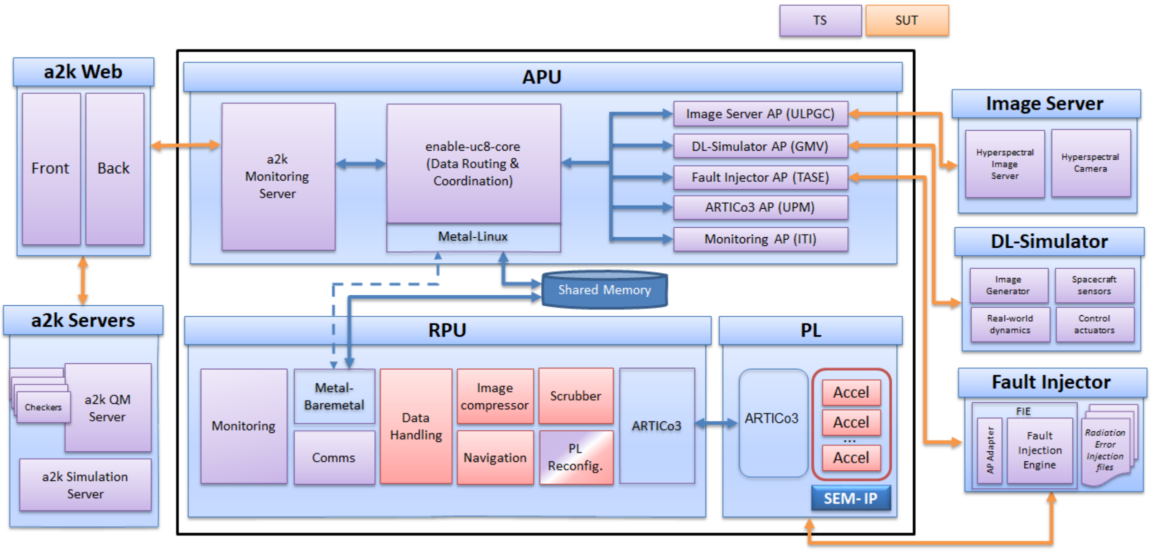


Figure 4: Test framework architecture

statistics of the SUT, of processing them, and of displaying the results in a unified interface.

This new service, integrated into the Test Framework (see section 4.2.1), has been made with the purpose (1) of finding system events not taken into account in the initial model, (2) of obtaining a better approximation of the system performance, and (3) of computing the observed temporal behaviour, defined by the Observed Worst-Case Execution Time and the Observed Worst-Case Response Time.

The monitoring tool allows the engineer to discard unfeasible temporal configurations of the SUT and to define the component implementations to be optimised. It also allows collecting application specific data, such as fault conditions or the use of shared resources, with the aim of validating the system executions.

An intrusive but small overhead software technique has been developed to collect the necessary data and to compute the temporal information of the test executions. A trace recorder component must be added to the SUT in order to be able to store the different temporal events and the time when they have been executed. The implementation complies with the Test Framework interfaces and with its internal communication protocol.

4.2.3 External applications

In order to build the overall test scenario, external applications are connected to the test framework using the APs interface. They must provide the necessary information to simulate the environment or to analyse the output data, that can be used in a close-loop simulation or to verify the system behaviour.

Fault Injector The possibility of injecting real faults in the FPGA, according to the expected failure obtained by analyzing the radiation environment, allows the evaluation of the behaviour of the designs installed in the SUT in the different stages of development. The fault injector permits the development of test set-ups adapted to the failures expected. This, in turn, reduces the number of visits to the BEAM, the testing time, and, in general, provides robustness against radiation to the design.

The Fault Injection Engine is a mix of the hardware on the actual device in which the engineer wants to inject faults and a software, which is capable of controlling this hardware. Verification and validation are essential steps in the development process of any autonomous system and as such represent key targets of ENABLE-S3.

Image server The on-board image compression algorithm, implemented in the exposed use case, must be connected to a real camera, provider of the hyperspectral images to be compressed. In order to emulate this behaviour, an hyperspectral image server is connected to the test framework via AP, working as an input sensor from the point of view of the SUT.

DL-Simulator The vision-based navigation algorithms integrated in the SUT needs, not only the hyperspectral images provided by the image server, but also the spacecraft sensors data, in order to execute the navigation filter and to manage the control actuators. For this reason, a specific space simulator (DL-Simulator) has been integrated in close-loop into the test system using the APs, providing the data of the emulated sensors to the SUT and using the actuator signals generated by the on-board navigation to manage the simulated spacecraft.

5 Concluding Remarks

The developments achieved in the aerospace use cases in ENABLE-S3 allow testing the applicability and suitability of COTS extensively. Moreover, evaluating their performance, by simulating the physical environment and assessing the safe and secure placing before testing under real conditions, reduces the cost of test campaigns and to acquire a greater knowledge about the behaviour of the system. This opens the possibility of new more flexible developments, with high performance and at a lower cost than using the traditional approach of aerospace.

Acknowledgements

This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL Joint Undertaking under grant agreement No 692455. This joint undertaking receives support from the European Unions HORIZON 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

References

[1] H. Winner and W. Wachenfeld, "Absicherung automatischen Fahrens, 6," *FAS-Tagung München, Munich*, vol. 9, 2013.

[2] G. Martin, "NewSpace: The emerging commercial space industry," tech. rep., NASA Ames Research Center, 2015.

[3] A. Leitner, D. Watzenig, and J. Ibanez-Guzman, *Validation and Verification of Automated Systems: Results of the ENABLE-S3 Project*. Springer International Publishing, 2019.

[4] L. Armesto Caride, A. Rodríguez, A. Pérez Garcia, S. Sáez, J. Valls, Y. Barrios, A. J. Sanchez Clemente, D. González Arjona, Á. J.-P. Herrera, and F. Veljković, *Reconfigurable Video Processor for Space*, pp. 231–249. Validation and Verification of Automated Systems: Results of the ENABLE-S3 Project, Springer International Publishing, 2019.

[5] A. Rodríguez, J. Valverde, J. Portilla, A. Otero, T. Riesgo, and E. de la Torre, "FPGA-Based High-Performance Embedded Systems for Adaptive Edge Computing in Cyber-Physical Systems: The ARTICo3 Framework," *Sensors*, vol. 18, no. 6, p. 1877, 2018.

[6] R. Guerra, M. Díaz, Y. Barrios, S. López, and R. Sarmiento, "A Hardware-Friendly Algorithm for the On-Board Compression of Hyperspectral Images," in *2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–5, IEEE, 2018.

[7] L. Sha, T. Abdelzaher, K.-E. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. K. Mok, "Real time scheduling theory: A historical perspective," *Real-time systems*, vol. 28, no. 2-3, pp. 101–155, 2004.

[8] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, vol. 24. Springer Science & Business Media, 2011.

[9] M. H. Klein, T. Ralya, B. Pollak, R. Obenza, and M. G. Harbour, *A Practitioner's Handbook for Real-time Analysis*. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[10] K. Tindell and J. Clark, "Holistic schedulability analysis for distributed hard real-time systems," *Microprocess. Microprogram.*, vol. 40, pp. 117–134, Apr. 1994.

[11] M. García-Gordillo, J. J. Valls, and S. Sáez, "Heterogeneous Runtime Monitoring for Real-Time Systems with art2kitekt," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 266–273, Sep. 2019.