A Model-driven development framework for highly Parallel and EneRgy-Efficient computation supporting multi-criteria optimisation

D1.4 Analysis of functional safety aspects on multi-criteria optimization and final release of the test bench suite

Version 1.0

Documentation Information

Contract Number	871669
Project Webpage	https://www.ampere-euproject.eu/
Contractual Deadline	30.09.2022
Dissemination Level	Public (PU)
Nature	R
Authors	Michael Pressler, Dirk Ziegenbein (BOS)
Contributors	Massimilano Polito (THALIT) Enkhtuvshin Janchivnyambuu (SYS)
Reviewer	Massimilano Polito (THALIT)
Keywords	safety, AMALTHEA, SLG

AMPERE project has received funding from the European Union's Horizon 2020 research and innovation programme under the agreement No 871669.

Change Log

Version	Description Change
V0.1	Initial version and structure
V0.2	Included partner contributions
V0.3	Included first review comments
V0.4	Version for final review
V1.0	Final version for submission

Table of Contents

Executive summary

This document describes Deliverable 1.4, related to WP1 *System Model Description and Use-cases* and targeting milestone MS3 (i.e., from M16 until M33, with a 6 months extension).

This deliverable corresponds to the final release of the test bench suite, upon which single- and multi-criterion optimization model transformations are applied. The tasks *Task 1.2: Generation of AMPERE test bench suite and use case preparation* as well as *Task 1.3: Functional safety and security* contribute to this deliverable.

Task 1.2, involving partners THALIT and BOS, develops a test bench suite, composed of the use cases as well as smaller and manageable kernels resembling a limited functionality of each use case which shall enable rapid evaluations of the AMPERE optimization framework.

Task 1.3, involving partners TRT, THALIT, BOS, EVI and SYS, evaluates the functional safety and security aspects required at each integrity level as defined in the relevant standards. The purpose is to ensure that the AMPERE ecosystem as well as the HW/SW platform complies with these requirements.

The second and final milestones of Tasks 1.2 and 1.3 have been carried out successfully, and all objectives of MS3 have been reached and are documented in this deliverable.

1 Introduction

The AMPERE project targets safety-critical applications and systems. Thus, safety-related requirements on the AMPERE ecosystem have been analyzed and documented. Furthermore, these requirements have been linked to the use case models which form the heart of the AMPERE Test Bench Suite.

This deliverable gives an update (in comparison to deliverable D1.2 [1]) on the activities and results with respect to the relevant functional safety aspects in Chapter 2 and discusses the final release of the AMPERE Test Bench Suite in Chapter 3. Since no specific security activities are required in AMPERE (see AMPERE Deliverable D1.2 [1]), this report does not cover security.

2 Functional Safety Aspects

This chapter summarizes the safety-related activities and results which have been carried out by the AMPERE consortium since the submission of AMPERE deliverable D1.2 [1]. In the following Section 2.1, more insight into the safety norms of the railway domain as well as into the results of the Preliminary Hazard Analysis (PHA) performed by THALIT for the ODAS use case including some new use case requirements in Subsection 2.1.2.2 are reported. Section 2.2 shows how the safety-related requirements of the automotive domain have been realized in the PCC use case and Section 2.3 points to deliverables describing the safety-related capabilities of the AMPERE SW platform.

2.1 Railway Use Case

2.1.1 Introduction

Most of railway safety rules are addressed in the following CENELEC standards:

- EN50126;
- EN50128;
- EN50129;
- EN50159.

Below some more information will be presented on the scope of work of safety standards (with a particular emphasis on EN50126 and EN50128) with an analysis of their impact on AMPERE ecosystem.

2.1.1.1 EN50126

CENELEC Standard EN50126 covers all topics related to the RAMS process (Reliability, Availability, Maintainability and Safety). The RAM part is not in the scope of this project so the description of this standard will focus on the Safety part.

EN50126 describes a process framework to consistently manage safety concerns using methods and tools that don't depend on the system/subsystem technology. This framework is called "the hourglass model" and is a simplified approach to help clarify some issues (see Figure 1).

Safety aspects are identified and managed in two phases: risk assessment and hazard control.

The "risk assessment" phase is the equivalent of a system hazard analysis. It starts from the system definition, works in a railway duty holder/operator perspective, and it's main focus is on identifying high level hazards. No technical details are considered at this stage and the system is analysed as a *black box*. Hazard mitigations are identified and they become part of the System Requirement Specification.

The "hazard control" phase consists of a more comprehensive and thorough risk analysis and is equivalent to a subsystem hazard analysis. The analysis is refined and focused on hazards and their causes, using detailed information derived from the technical solutions.

EN50126 standard also provides methods to derive the Safety Integrity Levels (SIL) for safety related functions.

Besides risks assessment and hazard control, the standard also sets constraint on company organization. One of the goal of these organizational requirements is enforcing independence among roles. Independence is needed to reduce the probability of systematic failures, i.e. failures depending on poor design or poor Verification & Validation. An example of an organisational breakdown structure compliant to EN50126 is reported in Figure 2.

To put all above in the context of AMPERE ecosystem, we can conclude that three kind of safety requirements are expected to be adhered to:

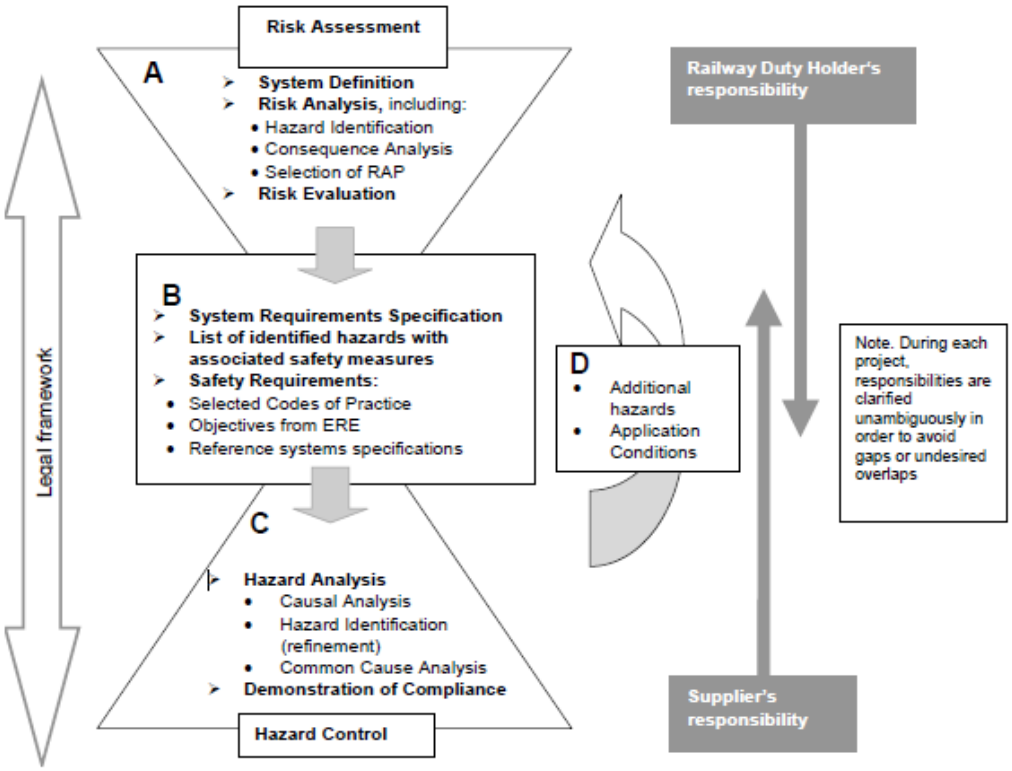


Figure 1: The Hourglass model.

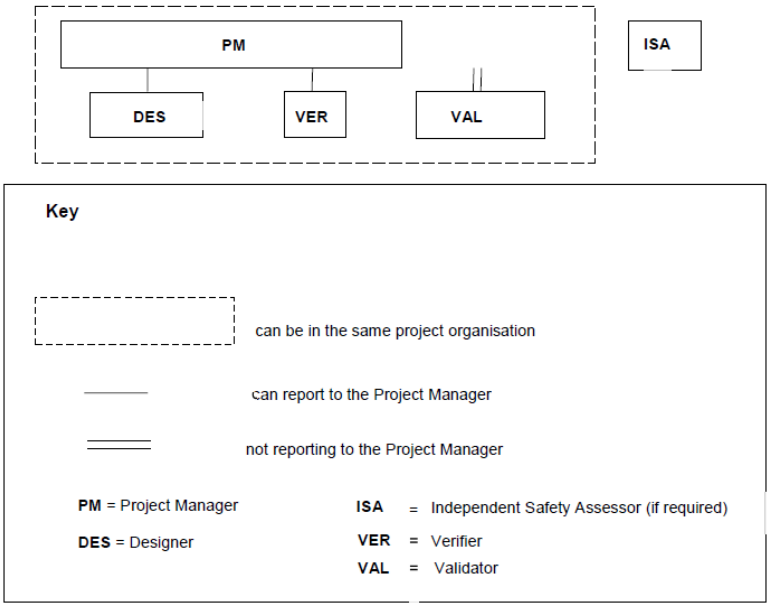


Figure 2: Independence of Roles in the project life-cycle (according to EN50126).

- requirements on the system under construction (the implemented system must be safely operated and maintained);
- requirement on the AMPERE ecosystem (i.e. a designer must find guidance in the tools to operate according to safety standards);
- requirement on the organization using AMPERE ecosystem (see Figure 2)

2.1.1.2 EN50128

EN50128 provides a process (with organisational structure and division of responsibility) and methodologies for development, deployment, verification, validation and maintenance of any safety-related software for the railway industries having been allocated a given software integrity level.

This standard has been built around the key concept of Safety Integrity Level (SIL) and gives guidance on how a SW must be designed, implemented, validated and maintained to be compliant to a given SIL level.

Five SIL levels have been defined by this standard:

- ***SIL0*** (the minimum SIL level for a SW that must be installed in a railway environment);
- ***SIL1***;
- ***SIL2***;
- ***SIL3***;
- ***SIL4***.

EN50128 specifies SW requirements both in terms of process (organizational requirements, documents to be provided, etc.) and in terms of techniques that can/can't be used in the development and testing of SW of a given SIL level.

Organizational requirements specify the kind of organisational breakdown structure that must be used according to the SIL level of the SW under development. As can be seen in Figure 3, a more detailed description of the organisational structure is given compared to the one presented in EN50126 (see Figure 2).

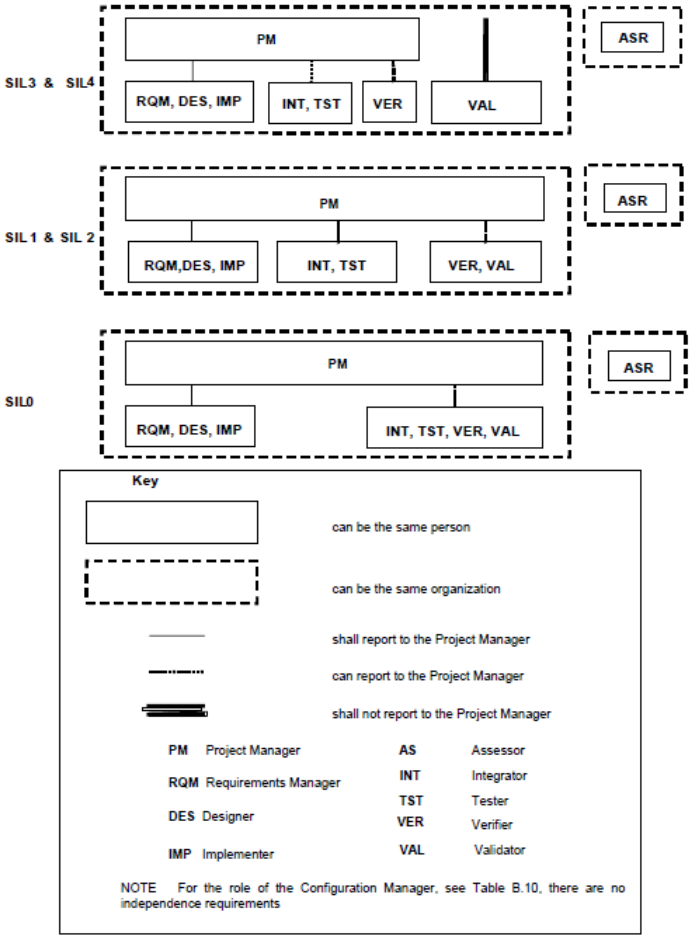


Figure 3: Independence of Roles in the project life-cycle (according to EN50128).

The emphasis on independence of roles is strictly linked to the SIL level of the final SW to be released. For the lower level integrity level (SIL0) project technicians can be spread on two teams and all people working on the SW design, implementation, verification and validation can report to the same person (the project manager - PM).

By raising the safety requirement to SIL1, SIL2 and SIL3 levels more and more independent teams must be organised. At the highest safety level (SIL4) not only people must be split in four teams but the validation team must be completely independent from the project manager.

EN50128 also put an emphasis on document production and for each project phase a given set of documents to be delivered is defined.

Besides organisational and documentation requirements, EN50128 requirements involve also the SW production phases. Some examples of SW requirements related to various SW life-cycle phases are given in Figure 4, Figure 5, Figure 6 and Figure 7 .

The following legend must be used to understand what techniques are required for each SIL level.

- ***'M'*** the corresponding technique is mandatory for the given SIL level;
- ***'HR'*** the corresponding technique is highly recommended for the given SIL level;
- ***'R'*** the corresponding technique is recommended for the given SIL level;
- ***'-'*** the designer is free to decide whether to follow or not the technique;
- ***'NR'*** the corresponding technique is not recommended for the given SIL level, its use must be avoided or its rationale justified.

TECHNIQUE/MEASURE	Ref	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Formal Methods (based on a mathematical approach)	D.28	-	R	R	HR	HR
2. Modelling	Table A.17	R	R	R	HR	HR
3. Structured methodology	D.52	R	R	R	HR	HR
4. Decision Tables	D.13	R	R	R	HR	HR
Requirements: 1) The Software Requirements Specification shall include a description of the problem in natural language and any necessary formal or semiformal notation. 2) The table reflects additional requirements for defining the specification clearly and precisely. One or more of these techniques shall be selected to satisfy the Software Safety Integrity Level being used.						

Figure 4: SW requirements specification.

2.1.1.3 EN50129

EN50129 is focused on the evidence to be presented for safety-related systems acceptance. The concept of a Safety Case is used as a framework for a systematic documented approach to quality management, safety management, functional and technical safety, safety acceptance and approval. This standard also contains details on SIL use in safety-related systems for railway applications.

2.1.1.4 EN50159

EN50159 has also been taken into consideration in our analysis as it deals with safety related communication between safety-related equipment using a transmission system in railway applications.

TECHNIQUE/MEASURE	Ref	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Defensive Programming	D.14	-	HR	HR	HR	HR
2. Fault Detection & Diagnosis	D.26	-	R	R	HR	HR
3. Error Correcting Codes	D.19	-	-	-	-	-
4. Error Detecting Codes	D.19	-	R	R	HR	HR
5. Failure Assertion Programming	D.24	-	R	R	HR	HR
6. Safety Bag Techniques	D.47	-	R	R	R	R
7. Diverse Programming	D.16	-	R	R	HR	HR
8. Recovery Block	D.44	-	R	R	R	R
9. Backward Recovery	D.5	-	NR	NR	NR	NR
10. Forward Recovery	D.30	-	NR	NR	NR	NR
11. Retry Fault Recovery Mechanisms	D.46	-	R	R	R	R
12. Memorising Executed Cases	D.36	-	R	R	HR	HR
13. Artificial Intelligence – Fault Correction	D.1	-	NR	NR	NR	NR
14. Dynamic Reconfiguration of software	D.17	-	NR	NR	NR	NR
15. Software Error Effect Analysis	D.25	-	R	R	HR	HR
16. Graceful Degradation	D.31	-	R	R	HR	HR
17. Information Hiding	D.33	-	-	-	-	-
18. Information Encapsulation	D.33	R	HR	HR	HR	HR
19. Fully Defined Interface	D.38	HR	HR	HR	M	M
20. Formal Methods	D.28	-	R	R	HR	HR
21. Modelling	Table A.17	R	R	R	HR	HR

Figure 5: SW architecture specification.

2.1.2 Safety Requirements

A Preliminary Hazard Analysis (PHA) has been performed starting from the safety function that characterises the system, i.e. detection and avoidance of track obstacles.

A number of hazards with their countermeasures have been identified. These countermeasures constitute a set of safety requirements.

Part of these safety requirements have been mapped to a set of higher-level mitigations to be used as safety requirements insisting on AMPERE design tool (i.e. AMPERE must satisfy some requirements to be able to design safety related systems), see section 2.1.2.1, while others have been kept as safety requirements for the use case under analysis, see section 2.1.2.2.

2.1.2.1 AMPERE safety requirements

As a result of PHA the following safety requirements insisting on the AMPERE ecosystem have been identified.

- ***[SYS-ODAS-REQ-200]**: AMPERE items implementing safety critical aspects must be managed (e.g., designed, implemented, tested, etc.) according to EN50126.*
- ***[SYS-ODAS-REQ-201]**: Code generation tools must consider EN50128 constraints when generating code for safety related design items.*
- ***[SYS-ODAS-REQ-202]**: If components are executed in an environment in which computing platforms are connected through a local interconnection network, the communication system must comply to EN50159.*
- ***[SYS-ODAS-REQ-203]**: Safety related SW modules and their corresponding configuration data can only be offloaded to hardware acceleration devices having the required Safety Integrity Level (SIL).*

TECHNIQUE/MEASURE	Ref	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Formal Methods	D.28	-	R	R	HR	HR
2. Modelling	Table A.17	R	HR	HR	HR	HR
3. Structured methodology	D.52	R	HR	HR	HR	HR
4. Modular Approach	D.38	HR	M	M	M	M
5. Components	Table A.20	HR	HR	HR	HR	HR
6. Design and Coding Standards	Table A.12	HR	HR	HR	M	M
7. Analysable Programs	D.2	HR	HR	HR	HR	HR
8. Strongly Typed Programming Language	D.49	R	HR	HR	HR	HR
9. Structured Programming	D.53	R	HR	HR	HR	HR
10. Programming Language	Table A.15	R	HR	HR	HR	HR
11. Language Subset	D.35	-	-	-	HR	HR
12. Object Oriented Programming	Table A.22 D.57	R	R	R	R	R
13. Procedural programming	D.60	R	HR	HR	HR	HR
14. Metaprogramming	D.59	R	R	R	R	R

Figure 6: SW design and implementation specification.

TECHNIQUE/MEASURE	Ref	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Formal Proof	D.29	-	R	R	HR	HR
2. Static Analysis	Table A.19	-	HR	HR	HR	HR
3. Dynamic Analysis and Testing	Table A.13	-	HR	HR	HR	HR
4. Metrics	D.37	-	R	R	R	R
5. Traceability	D.58	R	HR	HR	M	M
6. Software Error Effect Analysis	D.25	-	R	R	HR	HR
7. Test Coverage for code	Table A.21	R	HR	HR	HR	HR
8. Functional/ Black-box Testing	Table A.14	HR	HR	HR	M	M
9. Performance Testing	Table A.18	-	HR	HR	HR	HR
10. Interface Testing	D.34	HR	HR	HR	HR	HR

Figure 7: SW verification and testing specification.

- ***[SYS-ODAS-REQ-204]**: A design item attribute "safety related" must be available to designers to identify design items involved in implementing safety functions (safety related design items).*

Some of above safety requirements are directly impacting the use case model and they have to be managed at AMPERE design level (by adding information to the model and/or by extending tool functionalities). Some others can't be managed in the model or in the tool and will be transferred to third parties (AMPERE tool users) to be implemented.

Safety requirements to be implemented in the model/tool:

- ***[SYS-ODAS-REQ-200]**: AMPERE items implementing safety critical aspects must be managed (e.g., designed, implemented, tested, etc.) according to EN50126.*

- ***[SYS-ODAS-REQ-201]***: *Code generation tools must consider EN50128 constraints when generating code for safety related design items.*
- ***[SYS-ODAS-REQ-204]***: *A design item attribute "safety related" must be available to designers to identify design items involved in implementing safety functions (safety related design items).*

According to requirements [SYS-ODAS-REQ-200] and [SYS-ODAS-REQ-204] Capella model has been updated and tag "Safety Related" has been added to ODAS modules implementing safety functions.

Exported safety requirements:

- ***[SYS-ODAS-REQ-201]***: Code generation tools must consider EN50128 constraints when generating code for safety related design items.
- ***[SYS-ODAS-REQ-202]***: If components are executed in an environment in which computing platforms are connected through a local interconnection network, the communication system must comply to EN50159.
- ***[SYS-ODAS-REQ-203]***: Safety related SW modules and their corresponding configuration data can only be offloaded to hardware acceleration devices having the required Safety Integrity Level (SIL).

Above safety requirements can't be directly implemented in tools under THALIT/TRT scope of work and must be exported to third parties managing code generation, network communication and hardware allocation.

Please note that safety requirement [SYS-ODAS-REQ-201] appears both in the list of requirements managed in the model and in the list of requirements exported to third parties. A short explanation follows.

Requirement [SYS-ODAS-REQ-201] can't be totally implemented adding information to the Amalthea model as it is also related to the way this information is managed in tools following Amalthea in the tools chain (ex. code generating tools). As already done for requirements [SYS-ODAS-REQ-200] and [SYS-ODAS-REQ-204] above, Amalthea model has been updated and tags "SILx" have been added to the model to highlight safety related Amalthea items. However, tools following Amalthea in the tool chain must also be able to use the information added to the model to have the requirement correctly implemented.

More information on modification added to AMPERE tools (i.e. DSML extensions) to manage safety requirement can be found in AMPERE Deliverable D1.5 [2].

2.1.2.2 Use case safety requirements

As a result of PHA the following use case safety requirements have been identified:

- ***[SYS-ODAS-REQ-205]***: ODAS system shall perform a periodical vitality test ensuring the integrity of the sensors in charge to acquire the relative speed of obstacles (i.e. RADAR).
- ***[SYS-ODAS-REQ-206]***: ODAS system shall perform a periodical vitality test ensuring the integrity of the sensors in charge to acquire the relative position of obstacles (i.e. LIDAR).
- ***[SYS-ODAS-REQ-207]***: In the event ODAS system recognises a safety related issue, it shall react raising an alarm to the driver or by applying the tram brakes.
- ***[SYS-ODAS-REQ-208]***: ODAS system shall be based on a HW platform with a 2oo2 architecture.
- ***[SYS-ODAS-REQ-209]***: ODAS system safety related items must be SIL4.

Requirements from [SYS-ODAS-REQ-205] to [SYS-ODAS-REQ-207] are functional requirements and as such won't be developed any more as they are out of AMPERE scope of work.

Requirements [SYS-ODAS-REQ-208] and [SYS-ODAS-REQ-209] can be addressed through a hardware redundancy. Section 2.1.3 gives some details on a possible way to implement HW redundancy based on a "2oo2" architecture.

2.1.3 2oo2 Hardware architecture

2oo2 (pronounced "two-out-of-two") architectures are commonly used in systems where safety plays an important role: when a high level of safety must be reached a redundant hardware is often used.

A 2oo2 solution is based on the assumption (that can be mathematically proven) that the failure rate of a system consisting of two hardware working in parallel to perform the same function is lower than that of a single system performing the same function.

In a 2oo2 architecture, inputs coming from field devices are sent to two CPUs running the same application. The result of their calculations is checked for consistencies. If both CPUs agree on how to drive the output pins (i.e. both agree on what commands must be sent to field devices) the corresponding outputs are asserted. If an inconsistency is found between CPU results the system is stopped and taken to a "safe state" (i.e. a condition that continues to preserve safety).

This way, the risk of having a single point of failure is avoided and the safety of the system is greatly improved.

2.2 Automotive Use Case

In this section, we show how the safety-aspect related requirements defined in AMPERE Deliverable D1.2 [1] have been realized in the PCC use case AMALTHEA model. By giving concrete examples we show the link between the safety-related requirements and the use case, addressing the recommendation of the last project review.

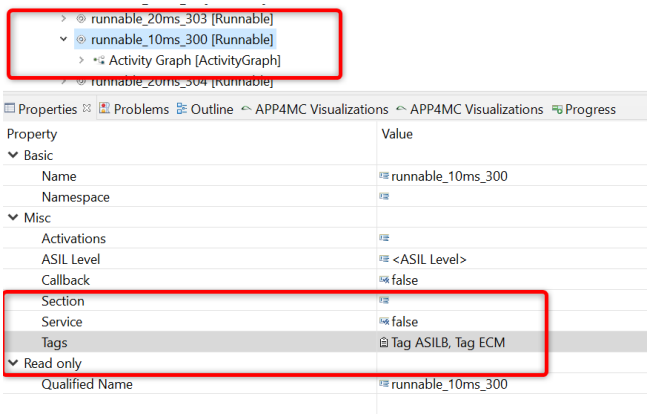


Figure 8: AMALTHEA tags specifies for each runnable its ASIL and the corresponding application

We use AMALTHEA tags for defining which runnables belong to which application as required by *[SYS-PCC-REQ-200]*. Figure 8 shows that *runnable_10ms_300* belongs to application *ECM*. This information will be used by the AMPERE tool chain to configure the temporal and spatial isolation mechanisms of the HW/SW platform to prevent fault propagation between different applications.

AMALTHEA tags are also used to assign an Automotive Safety Integrity Level (ASIL) to HW and SW model elements as required by *[SYS-PCC-REQ-201]* and *[SYS-PCC-REQ-202]*. Please note that AMALTHEA has an attribute *ASIL Level* fur runnables which we do not use as it does not support the specification of decomposed ASIL which we require for the ACC application of the PCC use case. Figure 8 shows that *runnable_10ms_300* is contributing to a safety goal with ASIL B. Similarly, Figure 9 shows that processor *CortexA53* has not been certified to a specific ASIL but has been developed according to regular quality management QM. This information is used by the AMPERE tool chain to ensure that ASIL X SW elements will be deployed on HW elements with ASIL $Y \geq X$ as well as that SW elements with an ASIL higher than or equal ASIL B will be executed in lockstep, either realized by redundant SW execution or by allocating them to a HW lockstep core as provided e.g. by the Infineon AURIX TriCore microcontroller family. See AMPERE Deliverable D1.2 [1] for more details.

The runnable *run81* of the adaptive Cruise Control (ACC) application is an example how two diverse implementations *run81 and run81_div* and a runnable separation constraint (as required by *[SYS-PCC-REQ-203]*) have been used to lower their ASIL from B to A(B). Figure 10 shows the ASIL tags of the runnables as well as the runnable separation constraint. The separation constraint clusters runnables that are not allowed to run on the same cores into two entity groups. The AMPERE tool chain will use this information to deploy the two runnables to different cores.

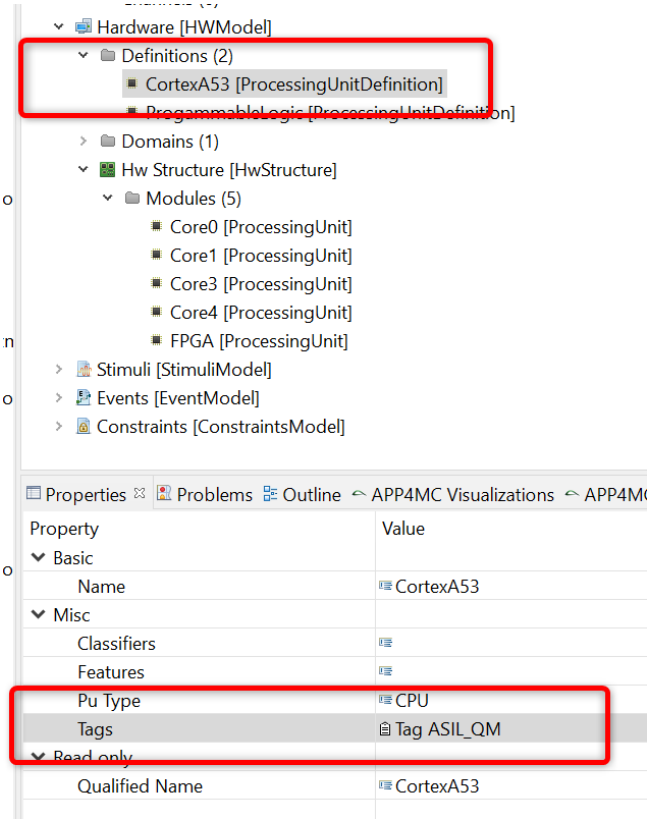
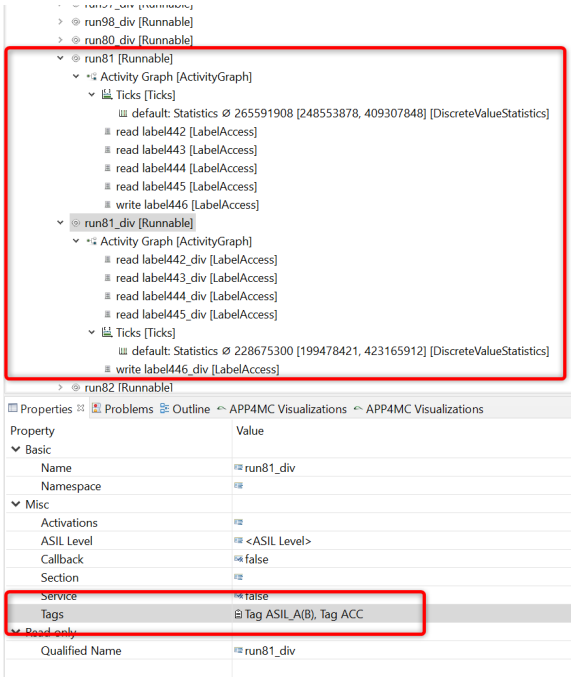


Figure 9: AMALTHEA tags specify the ASIL of a HW element

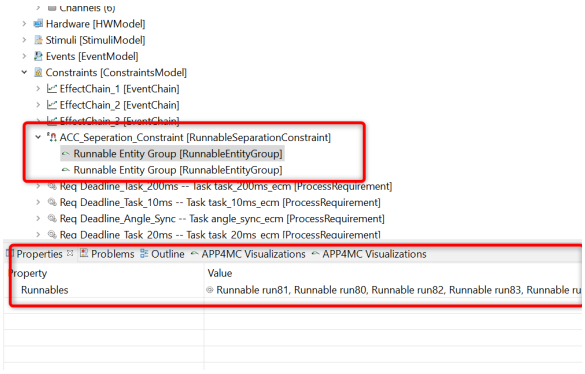
A complete description of the PCC use case model is contained in AMPERE deliverable D1.5 [2].

2.3 Safety Aspects of the SW Platform

PikeOS real-time operating system is designed and developed for safety and security-critical systems based on a separation kernel approach. Its separation kernel has a capability for a hypervisor with multiple partitions for many other operating systems and applications developed using user-level libraries, such as OpenMP. For detailed information, please refer to AMPERE deliverables D4.1 [3], D5.2 [4], and D4.3 [5].



(a) Diverse Runnables



(b) AMALTHEA runnable separation constraints prevent the allocation of two runnables to the same core

Figure 10: ASIL Decomposition with Runnable Separation

3 AMPERE Test Bench Suite

In this chapter, we will discuss the AMPERE test bench suite. Instead of implementing a fixed suite of test models, AMPERE developed and released a synthetic load generator (SLG). This way, each project partners is enabled to generate its own synthetic kernels of arbitrary complexity from AMALTHEA models in order to test their methods and tools. The extensions of the SLG beyond the state reported in AMPERE deliverable D1.2 [1] are described in Section 3.1.

The models and code of the ODAS and PCC use cases, another part of the test bench suite, are described in detail in AMPERE Deliverable D1.5 [2] in order to avoid repetition. The PCC use-case model can be found in the AMPERE repository [6].

3.1 Synthetic Load Generator

3.1.1 Execution Specializations

One of AMPERE's main goals is the exploitation of heterogeneous high-performance embedded hardware platforms. To cope with the modelling challenges we extended the Amalthea meta-model with the capabilities to model different implementation specializations. The detailed description of this extension is described in the Deliverable D1.5 [2]. In Figure 11 the runnable *computation* provides two implementation specializations. One for *host_omp* and one for *device_omp*. Each implementation is modelled in Amalthea with their data access and timing properties. Depending on the context set by the calling task or runnable the specialization is executed. In this example task [pointcloud2_to_image_device] calls the *device_omp* implementation.

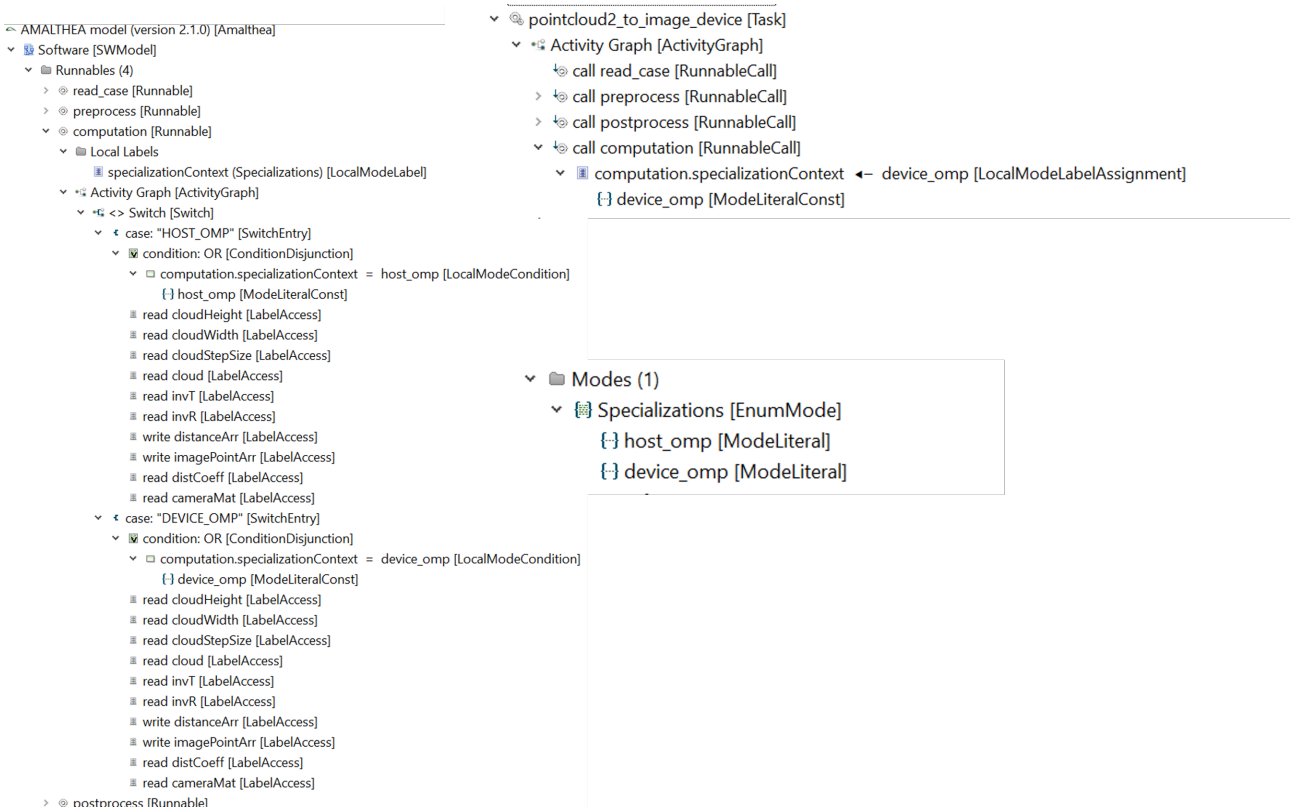


Figure 11: Modelling of Execution Specializations

If the decision is set in the call context this will be generated by the SLG. If the decision which specialization to

execute is empty, the Multi-Criteria Optimization identifies the best execution specialization for each runnable in each call context. The decision depends on the optimization goal, e.g. to optimize for high performance or for reduced energy consumption. This extension is implemented in the Synthetic Load Generator [7] to generate the implementation specializations as an input to the analysis tools.

3.1.2 ROS2/micro-ROS

We extended the SLG to cover the automatic generation of micro-ROS code. micro-ROS [8] was implemented to bridge the gap between resource constraint µControllers and performant µProcessors to build applications based on ROS2. The structure of the SLG repository is depicted in Figure 12. All boxes with the Eclipse logo are open source and available in the tools repository [7] of APP4MC. The model transformation framework provides a general infrastructure to implement model-to-model and model-to-text transformation with mechanisms to e.g. provide dynamic code injections. On top of this framework the basic SLG is implemented. The M2T plug-in provides the general implementation how to generate executable code from Amalthea models. The Linux, ROS2/micro-ROS, and AUTOSAR Adaptive plug-ins extend the basic SLG to provide further customization to support Posix threads, ROS2/micro-ROS nodes and communication primitives as well as specialized code for the ETAS AUTOSAR Adaptive implementation RTE-VRTE [9]. The latter is not publicly available.

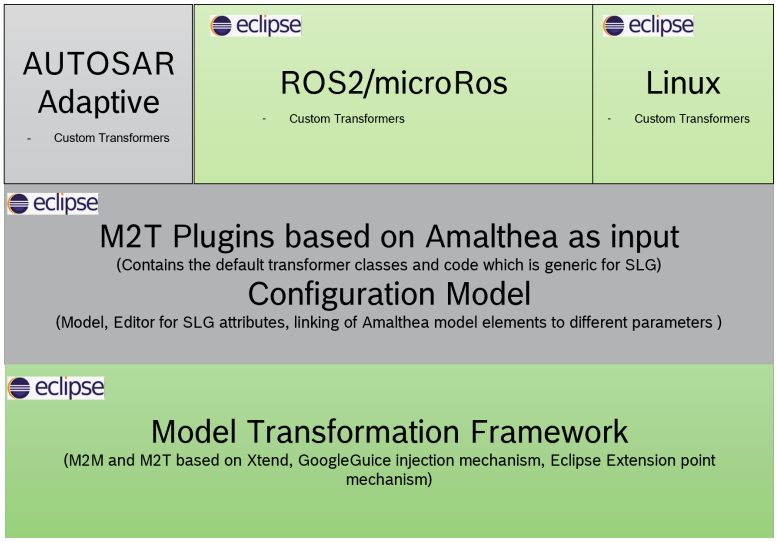


Figure 12: Synthetic Load Generator

In the Amalthea model the user can now choose, if a task will be part of a ROS2 or a micro-ROS node using TAGs as displayed in Figure 13. The user can define the TAG type to steer the code generation.

A ROS node is basically a process to perform some computation. The basic idea is, that a ROS node is independent and communicates with other ROS nodes via a pub/sub based communication mechanisms. In ROS nodes can subscribe and publish to so called topics which represent the data exchanged between nodes. In Amalthea we use channels to model these topics as illustrated on the left side in Figure 14 in the Amalthea model. Amalthea provides model elements that allow sending and receiving data from or to a channel.

While a publisher just writes data to a channel, a subscriber has different access patterns. It can access the channel after being periodically activated or the process can be asynchronously activated when a new data is pushed to the channel. This is depicted in Figure 14 where the micro-ROS databroker of the ECM is reacting to an event stimulus, that triggers the task every time a new *operationSetPoint* is received from the ACC.

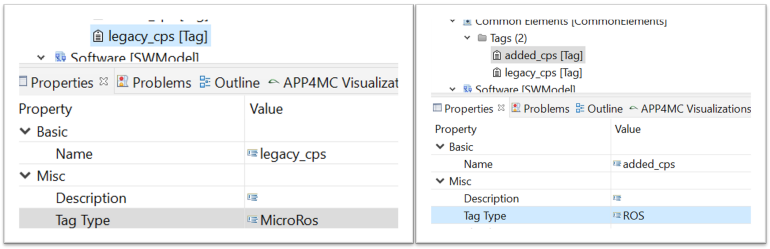


Figure 13: Modelling of ROS/micro-ROS nodes using Tags

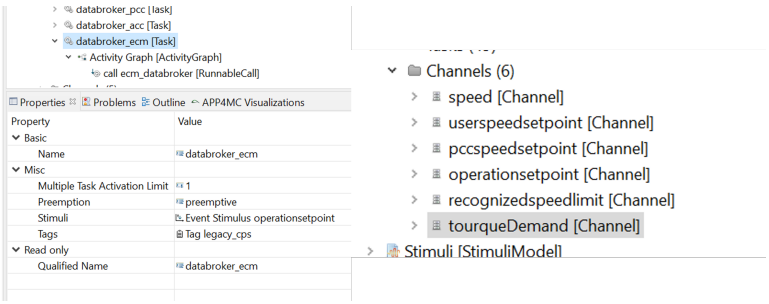


Figure 14: Modelling of ROS/micro-ROS topics and data driven activation

4 Conclusions

This document presented the contributions of WP1, related to safety. At this point, all safety-related analysis and modeling activities targeted towards the AMPERE ecosystem have been finalized.

Now, the final steps to be performed in the AMPERE project in the context of WP1, are related to the validation of the proposed DSML extensions and the optimization methodology using the two application use cases. The purpose of this use case based evaluation is to demonstrate that the AMPERE ecosystem is able to fulfill the safety-related requirements defined in this deliverable and its preceeding deliverables [10, 1].

5 References

[1] AMPERE, "D1.2 Analysis of functional safety aspects on single-criterion optimization and first release of the test bench suite," 2021.

[2] ——, "D1.5 Meta model-driven abstraction and model-driven extensions and use case enhancements," 2022.

[3] ——, "D4.1 Run-Time Architecture," 2020.

[4] ——, "D5.2 Single-criterion operating systems and hypervisor software," 2021.

[5] ——, "D4.3 Integrated run-time energy support, and predictability, segregation and resilience mechanisms," 2022.

[6] "PCC Use-Case Model," https://gitlab.bsc.es/ampere/ampere-project/-/tree/master/Software/PCC%20model%20in%20Amalthea/PCC_UseCase.

[7] "APP4MC Transformation Repository." [Online]. Available: https://git.eclipse.org/r/plugins/gitiles/app4mc/org.eclipse.app4mc.addon.transformation

[8] "micro-ROS." [Online]. Available: https://micro.ros.org/

[9] "ETAS RTA-VRTE." [Online]. Available: https://www.etas.com/de/portfolio/rta-vrte.php

[10] AMPERE, "D1.1 System models requirement and use case selection," 2021.