CPS Tool Evaluation

Deliverable ID:	D5.6 (D74) CPS Tool Evaluation
Version:	2022
Due Date:	September 30, 2022
Dissemination level:	Public

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

CPS4EU
WP 5 – D5.6 CPS Tool Evaluation

Document Manager:	Valery Morgenthaler		
Project Title:	Cyber Physical Systems for Europe		
Project Acronym:	CPS4EU		
Contract Number:	826276		
Project Coordinator:	VALEO		
WP Leader:	CEA		
Task:	T5.1 T5.2 T5.3 T5.4 T5.5	**Task Leader:**	ANSYS
Document ID:	D5.6	**Version:**	Rev1.0
Deliverable Title:	CPS Tools – Tool Evaluation	**Date:**	September 30, 2022
		Approved:	
Document Classification:	Public		

Approval Status

Prepared by:	Valery Morgenthaler (ANSYS) and Luis PALACIOS MEDINACELLI (CEA)
Approved by (WP Leader):	Réda NOUACER (CEA)
Approved by (Coordinator):	Philippe Gougeon (VALEO)
Approved by (TPM)	Etienne HAMELIN (CEA)

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Contributors

Name	Partner
Réda Nouacer, Morayo Adedjouma, Yves Lhuillier, Zakaria Chihani, Palacios Luis, François Terrier, Chokri Mraidha, François Bobot, Marwa Zeroual, Gilles Mouchard	**CEA**
Valéry Morgenthaler	ANSYS
	INRIA
Philippe Fiani, Benjamin Thiron, Sahar Guermazi	SHERPA-Engineering
Siddhartha Gupta	TUC
Alexander Mages, Sebastian Ochs	TRUMPF
Noël Hagemann, Julia Rauscher, Bernhard Bauer	UnA
Antonio Ruiz-Alba, Miguel García Gordillo, Javier Coronel	ITI
Salim Chehida	UGA

Version History

Version#	Date	Reason for change	Released by
0.1	June 15, 2022	Template First Release	V. Morgenthaler
0.2	August 8, 2022	First Draft with all contributions	V. Morgenthaler
0.3	Sept. 21, 2022	Integration of missing inputs INRIA, TUC, ITI Consolidation of document	L. Palacios
1.0	Octo. 10, 2022	Review and minor fixes	R. NOUACER

Distribution List

Name	Company/Organization	Role / Title
Consortium	CPS4EU Consortium	n/a

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

TABLE OF CONTENTS

Deliverable D5.6 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

TABLES

FIGURES

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

EXECUTIVE SUMMARY

In the previous deliverable D5.5 (CPS Tool Final Implementation and Integration), we have demonstrated the feasibility of the loose interactions between the different partner's tools as defined in the deliverable D5.3.

Partners worked together under three clusters: (a) heterogeneous co-simulation cluster that demonstrates how Heterogeneous simulation components, generated from different tool eco-systems, can be integrated in distributed simulations environment. (b) scenario-based simulation cluster where it is demonstrated how it is important to define scenarios in a formalized way and how this help simplify the process of scenario development and make it accessible for different CPS to use scenarios for its safety assessment (c) Modelling and analysis of AI-Based systems where is demonstrated the feasibility of the integration of Knowledge Bases and Reasoning into Industrial tools, tackling specifically system design.

Partners then demonstrate results of the different integration workflows on two use cases defined at the beginning of the deliverable: (a) Hybrid vehicle use case, and (b) Drone use case.

This deliverable is dedicated to the evaluation and validation of the final version of tools integration. All the tool clusters will provide for this a final version of the work shown in the current deliverable D5.6.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

1 INTRODUCTION

1.1 PURPOSE

This deliverable D5.6 is dedicated for the evaluation and validation of the final version of tools integration.

1.2 SCOPE

This document covers tasks:

- T5.1: AI integration in CPS
- T5.2: Simulation for CPS
- T5.3: Trustworthy system engineering
- T5.4: Tool chain development and integration

1.3 LINK TO OTHER DOCUMENTS/TASKS

ID	Description
WP6	CPS Pre-integration
WP7	CPS Automotive
WP8	CPS Industry automation
WP9	CPS for other industrial sectors

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Definition / acronym / abbreviation	Description
PHEV	Plug-in hybrid electric vehicles
EMS	Energy Management System (EMS)
CPS	Cyber Physical system
AI	Artificial Intelligence
FMI	Functional Mock-up Interface standard
FMU	Functional Mock-up Unit
HLA	High Level Architecture standard
ADS	Automated Driving Systems
SDL	Scenario Definition Language
DSL	Domain-Specific Languages
SES	System Entity Structure
PES	Pruned Entity Structure
CTE	Cross-Track Error
OML	Ontology Model Language
OWL	Web Ontology Language

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

2 EVALUATION OF THE WORK DONE BY THE PARTNERS

2.1 CEA

2.1.1 Summary and overview

The CEA has been in charge of the research on integration of AI into the Cyber-Physical Systems lifecycle. This involves integration of systems and components carrying AI, the analysis and validation of with respect to the systems requirements and enabling AI driven tools for the design of CPS. A first identification of the tools and their possible interaction is represented in Figure 1. Our work is concerned by the modelling of the systems, their safety analysis, as well as their verification through formal methods and hybrid approaches.

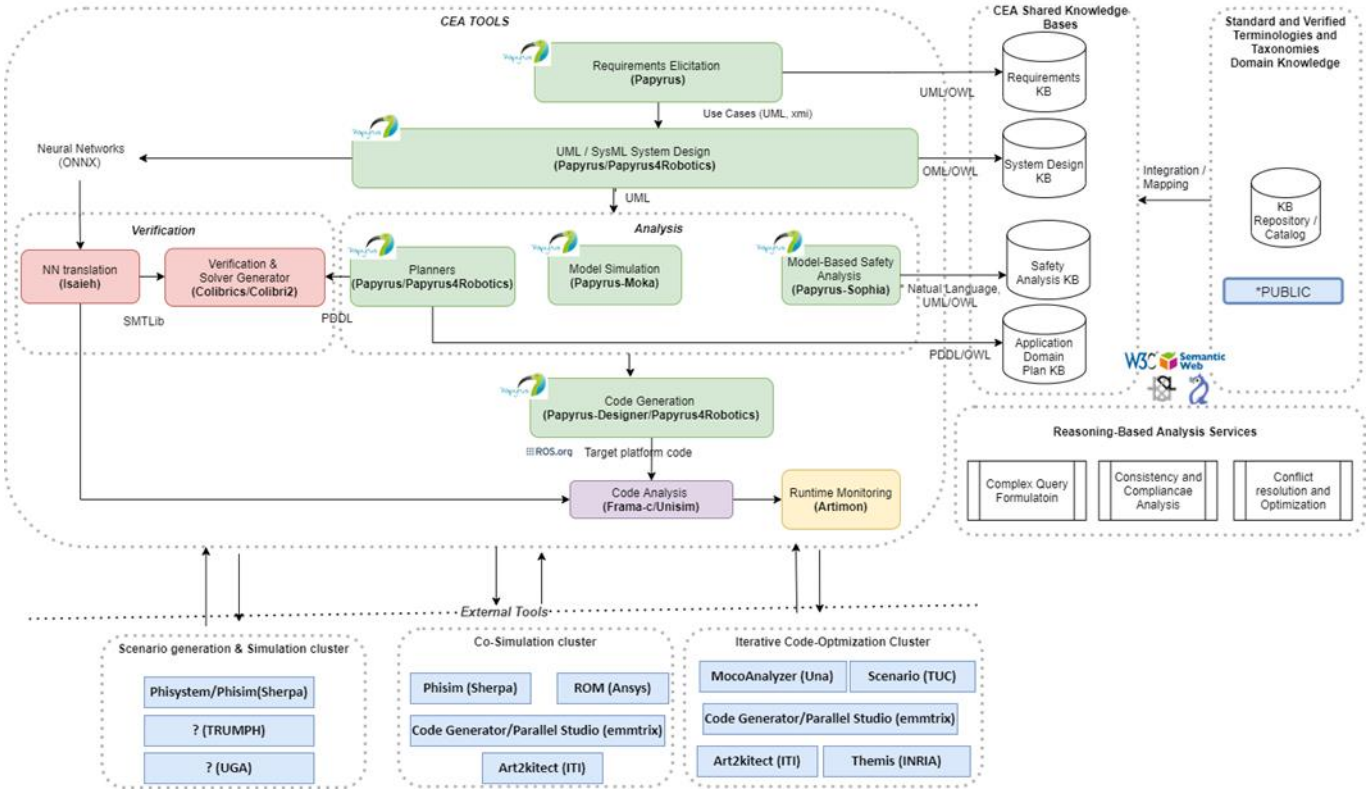


Figure 1: Overall view of proposed cluster.

2.1.1.1 Knowledge Based System Engineering

Of key importance in our tool, framework context is the ability to share the models used in each tool, to enable interaction and communication in large and highly heterogeneous environments. Current engineering tools are sophisticated and support rich expressiveness to describe the systems being modelled, but at the expense of requiring high expertise in the tool's specific representation language (i.e. UML/SysML). Moreover, new projects have to be built from scratch rebuilding structures and descriptions that are common to specific domains (e.g. Autonomous Systems). Despite the formalized nature of MBSE and the rich expressiveness of UML and SysML, current tooling environments have limited interpretation capabilities regarding the content and semantics of the system being modelled. The current design tools and the languages they rely on, provide benefits regarding traceability of the models, constraint checking and formalization, but at a syntactic level, whereas the semantics and machine-readable models are still out of the scope of these tools.

In the domain of Artificial Intelligence, the Knowledge Representation and Reasoning (KRR) domain is concerned with the formal representation of knowledge. This includes its semantics, the ability to share unambiguous representations among heterogeneous stakeholders and efficient machine-readable models over which we can reason. Thus the integration of KR and semantic technologies into MBSE, enables a system's model designed in a specific tool with a specific concern/point of view, to be shared in a larger ecosystem, that share the common semantics.

Regarding MBSE, it is important to highlight that the approach is not only intended to enable model sharing among heterogeneous tools and stakeholders in large ecosystems, but it also enables locally (i.e. within the tool) the ability

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

for reconfiguration of components (possibly carrying AI) in a system, while preserving the systems overall properties. This is possible thanks to the attached semantics and reasoning capabilities, since the tool can interpret what is being designed, and the constraints of the system can be expressed in terms of capabilities or skills (e.g. object detection or collision avoidance capabilities). One of the axes to achieve the interaction of these tools and the components reconfiguration relies on the ability to integrate domain specific knowledge into the tooling environment. This feature allows the tool-expert to use the domain knowledge in the engineering process; it offers documentation about the ontology and the mapping to/from the tools formalism; and if provides the capability to export the model from the tooling environment into a W3C compliant representation. This W3C representation of the systems model can be shared in a larger ecosystem, and can be evaluated against DL-Concepts, SPARQL queries and SWRL rules, among others.

2.1.1.2 Safety and Hazard Analysis

In the last decade, safety has become one of the most relevant concerns when designing AI-based systems as humans are sidelined progressively from the decision/control loop of intelligent and learning-enabled machines. Nevertheless, the safety of such systems is also a vast and complex subject considering the various types of AI technologies and the multiple applications domains where they can be deployed, together with their related specificities.

In particular, we observe that there is a lack of common criteria and thresholds for safety evaluation of AI-based systems. In traditional control systems, deductive inference logically links basic safety principles to implementation. However, the AI-based system of nowadays integrate more and more ML-based inductive inference to achieve extensive gains in autonomy. Inductive inference can yield excellent performance, in nominal conditions, but it may not produce semantically understandable rules, as it rather finds correlations and classification rules within training data. Validating such inductive learning is tough, due to our inability to collect an epistemologically sufficient quantity of empirical data to ensure correctness. The challenge is about ensuring that the overall scenario space of the system is correctly covered by safety relevant appropriate means, regarding the infinite continuum of complex environmental conditions and the non-deterministic nature of the embedded AI algorithms that the systems may be built on.

In addition, the implementation of AI systems create emerging categories of hazardous events. The system must deploy algorithms to identify and properly react to complex, rather unforeseen, situational scenarios and, on top of that, they should ensure negligible likelihood of occurrence for critical hazardous events and malfunction. Referred hazardous events exhibit an increased risk level due to the machine overtaking over former human based activities. Along with more autonomy, the transfer of duties to AI-based algorithms implies no further human interaction as safety barrier in case of hazards.

The numerous specificities of AI-based systems as referred above make the current engineering methods barely applicable for their development and assurance. The most widely used methods across industries are not anymore good candidates to cope with the heuristic nature of AI components while AI components may provoke accident without any internal failure or defect but according to its environment sensing and interpretation. The challenge then lies in defining a body of "desired" or "acceptable" behaviours for AI-based systems. The main safety activities affected by this particularity of the AI based systems is the hazard analysis at the concept phase of the system.

We develop a risk based approach that enable to identify and to assess risks to which such AI-based systems may be exposed and derive appropriate safety principles and mitigation measures. The approach focuses on the design-stage of the CPS development, with some early propositions for the operational phase. Of upmost importance, the safety methods must ensure that any scenario that may have happened in operational time will be properly handled by the AI system - within defined safety margins. To address the lack of specification of the AI application, we built a method for Operational Design Domain (ODD) definition. The ODD is used to define a situations catalogue for the system. This ODD, through expert knowledge conveyed in ontology, capture all scenarios in which the system must be designed to operate properly taking into account the operating limitations.

The ODD definition serves as input for the hazard analysis, more specifically for the identification of critical scenarios that may be issued by those AI systems. Our hazard analysis approach helps identify the conditions that will lead, if occurring in an operational situation, to an accident. For classical systems, such conditions are predefined in some hazard list and they are essentially based on known HW/SW failures. To cope with the novel AI-based systems, we consider also as initiating condition, any functional insufficiency, or human misuse to cope with the potential autonomous functions weaknesses. In overall, we address five main AI-induced potential hazard sources in the hazard identification: 1) Manoeuvre-based, 2) misuses behaviour, 3) functional insufficiencies, 4) components limitations and 5) triggering events related to environmental conditions. We propose a systematic approach for

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

deriving critical scenarios based on adapted and extended FMEA, STPA, HAZOP, SOTIF guidewords. We combine the keywords with operational scenarios specified from the ODD to identify hazardous events. The method must be applied at vehicle level (vehicle level functionality) but also at functional component level (e.g. perception, navigation, etc.) to derive critical scenarios as complete as possible.

The results of this design-stage analysis is used at runtime. Indeed, the ODD will serve to derive: 1) requirements on the architecture model of the system, based on the operating limitations identified from the relevant parameters; and the hazard analysis will serve to specify 2) safety constraints (including safety variables) needed to monitor the system at runtime - so that to avoid the system to exit the ODD, i.e. to detect during the operational phase any deviation from normal behaviour and apply necessary corrective actions.

2.1.1.3 Planning

The system's design specify, among other details, the components of a system. These components in turn provide the system as a whole with emergent capabilities like grasping, moving (on the ground or by flying), obstacle detection, obstacle avoidance, etc. Planning on the other hand, can be done relying formalisms like PDDL, where a domain and specific problems within the domain are fed into a solver to find sequences of actions, this sequence is then called a plan. These actions and their applicability are expressed in terms of facts and skills about the agents and systems that participate in the plan. A mission, for which a plan is searched, imposes constraints on both: the plan and the type of system capable of following the plan. An example of such constraints could be the payload capacity of a drone or obstacle detection capabilities of a system. These high-level skills, and their enforcement, are described in terms of predicates in a formal language (PDDL). We have explored and foreseen interaction between the planning specifications, the mission and the system's design. Papyrus for Robotics (P4R) enables the definition of planning for robotic applications. In P4R the design model is the entry point to generate a planning for the robot application, which also can embed safety constraints from a previous safety analysis. Both the system's design model and the planning –either augmented with safety properties or not - can be validated via simulation with Papyrus-Moka. The model is turned into an executable version using fUML or PSCS. Then, Papyrus Moka interprets the executable model for validating the system's high-level model visually and the planning respectively.

Some challenges to achieve this interaction, arise in the alignment of the plan's definitions a vocabulary with the viewpoint and granularity level of the system's model. Thus, this effort has been discontinued, to privilege the integration of ontologies into MBSE, and the export of these models as OWL. The rationale behind this decision is that enabling a tooling environment with these capabilities, eases future efforts in the integration of knowledge bases into specific tools, like P4R. Thus, this remains as further work.

2.1.1.4 Formal methods for verification, test, and safety by-design

NN properties can be assessed through formal verification (a global but relatively costly method) as well as through more localised property-based testing, where tests are generated automatically according to properties about the domain of applications. This two-pronged approach, which has been demonstrating its effectiveness in "traditional" (i.e., non-AI) software for decades, remains relevant for the paradigm shift brought by NN. Indeed, the long amassed experience of the Formal Methods (FM) community dedicated to human-written software has several guidelines that have proven essential. Chief among them is the necessity of a diversified tool belt, with various tools and methods tackling different aspects of safety.

NN is experiencing an outstanding growth in the number and the maturity of tools dedicated to evaluating their properties with rigorous, principled methods based on solid scientific bedrock. History taught us that this diversity could greatly benefit from a bridging platform that can relate the different tools and encourage synergy between their complementary features. CEA teams have dedicated part of their efforts in this project to achieving such a goal.

However, the fashionable focus on NN should not obfuscate the persistent relevance of other AI-paradigms, among which symbolic AI, such as constraint solvers and expert systems, remain the most prominent representatives. In this field, constraint programming (CP) software has a dual utilization. The first is that, like any prover, it can be used as a FM tool to help verify properties about other software. The second is that it can also be used, itself, as the subject of a validation process, including through FM. Indeed, CP tools are used in safety-critical settings (defence, e-Commerce, aviation, etc.), and as such, their validation can be paramount in the overall trust of the systems which use them. In this article, we focus on this second case and propose an answer to the question "how can we develop a modular and safe by-design constraint solver".

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

2.1.2 KPIs

Our developed technologies focus on addressing the Project's technical KPI 2.1 "Successful development of CPS modelling, simulation and verification tools to support the multi-purpose PIARCHs" (Figure 2 and Figure 3).

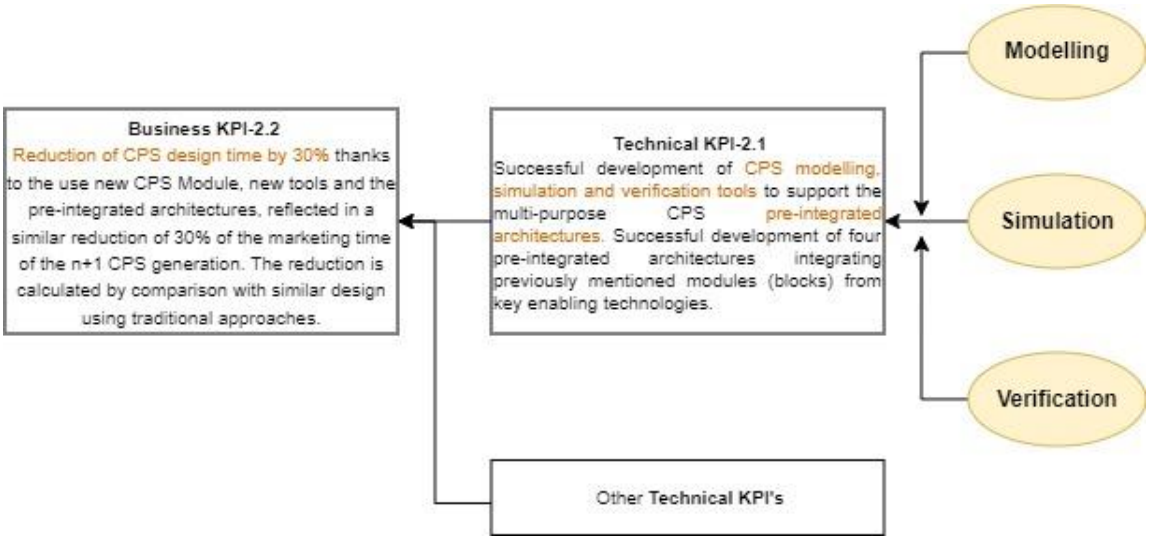


Figure 2: Business and Technical KPIs defined in D.5.2

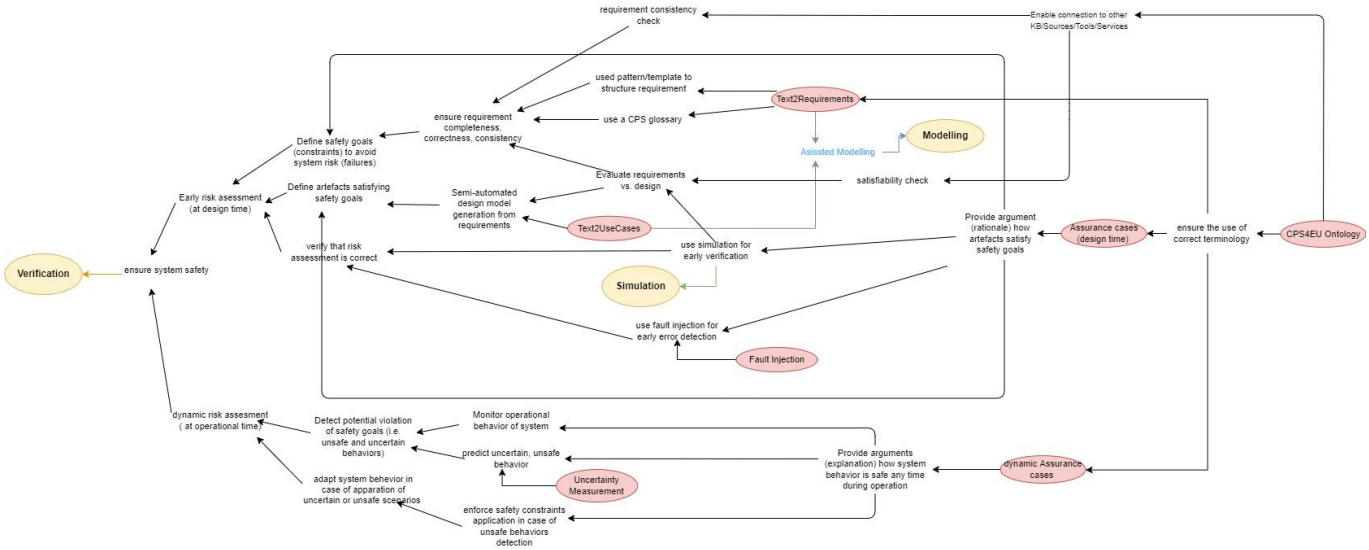


Figure 3: High-level technical KPIs

Figure 3 provides a roadmap for the interaction of different specific goals, and how they contribute to the main cluster's goal: to ensure systems safety. These goals have guided the efforts and the main interactions between different domains. While we have privileged the work related to system design, safety & risk analysis, scenario definition, simulation and verification; other areas such as: assurance cases, text2usecases, text2requirements, remain as further work and as potential candidates for future interactions.

Table 1 establishes a set of Key Performance Indicators, that contribute to our afore mentioned goals (see Figure 3). In Table 1, on the left, the more general KPIs related to KPI-2.1 in Figure 3. This KPI is further specialized in more detailed questions/objectives in the next columns. Next, a set of proposed criterion and their details are presented. Finally, on the right of the table the results and their success evaluation is provided.

2.1.2.1 Knowledge Based System Engineering

We have relied on semantic technologies and knowledge representation techniques to provide our tooling environments with tool-agnostic representations of the systems, that can be (semi) automatically integrated and exported from the tool. The goal of this interaction is to enable industry-wide standard vocabularies to be applied in system's design, hazard analysis, properties verification, etc. Such models can also be shared among heterogeneous tools and the represented knowledge modularized depending on the tools/stakeholders specific

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

needs, therefore it is a step towards interoperability and seamless share ability of models among heterogeneous stakeholders. This also implies that constraints from an external tool/source can be evaluated against the models created locally, increasing the reliability of the models, enabling further early-error detection and reducing design, engineering and implementation costs. Some tasks, like the direct integration of the designed models into simulation environments and restricting the designs based on constraints coming from planning, have been explored but not had a follow-through. This is in part because of lack of resources, and because the gap between the tools, their viewpoints and the use cases are not as aligned as to enable current interaction or a demonstrator for it. On the other hand, we have successfully completed the workflow for the integration of the KBs into System Design, targeting the UAVs domain (Drones) and the Papyrus for System Design tool, relying on the IEEE1872 standard (CORA) for autonomous systems. We have provided a methodology, best practices, a standardized domain specific ontology, a feasible technology stack, a demonstrator and a publication about the research.

2.1.2.2 Safety and Hazard Analysis

For AI based systems, one must develop means both at design stage and at operational stage to ensure safety of such systems. We have developed several components to address the safety of AI-based systems.

First, we target the lack of specification of AI based systems by proposing a ODD specification methodology to determine the scenario-space of such system. We also developed a safety analysis method that enable to identify specific deficiencies and faults other than classical HW/SW faults that the AI systems may exposed or be exposed to.

The above solutions enable to address safety at design stage. However, AI systems may face unpredicted situations at operation time. Since those situations were not analysed during development time, the system may not be equipped to adequately respond in front of them. Therefore, it is also important to define means to ensure the system will remain safe in any situation it may encounter at operation time. For that purpose, we rely on monitoring and safety enforcement rules components. The goal is to ensure that the system is able to detect an unknown scenario when it happens, and to execute dependable decisions in such case.

We provided tool support for hazard identification based on ODD and a runtime monitoring & safety enforcement simulation environment as demonstrators of our work. We also propose a generic framework that glue all the above components for a complete dynamic dependability management of autonomous systems. These research works lead to several publications.

2.1.2.3 Verification

We have developed a formally verified library of constraints, ColibriCS, that can be used to generate tailor-made symbolic AI components in several languages, including C, which is of great importance when targeting an embedded application. The modularity of ColibriCS allows for the separate definition and proof of several components (domains, propagations, labelling, etc.) and then their close to seamless aggregation following the needs of the application domain.

We have also developed the first version of CAISAR, the open-source platform for Characterizing AI Safety And Robustness. It is modular: several tools can be plugged in the platform with minimal effort through a flexible API. CAISAR is also diverse in its methods and its targets. In its methods: it integrate several formal methods, ranging from property-based testing (e.g. AIMOS, the AI Metamorphism Observing Software) to formal verification and reachability assessment (e.g. through the integration of PyRAT, the Python Reachability Assessment Tool). It is also diverse in its targets: we have developed CAISAR to be more general than just the fashionable NN: for example, it also integrates tools that target SVM (Support Vector Machine) such as SAVer.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Question / Objective			Proposed success criteria/criterion	Detail	RESULTS
Knowledge BaKnowledge Based System Engineeringsed System Engineering					
How to enable a tooling environement with formal shared semantics, where models and constraints can be unabiguously shared among tools and stakeholders?	**How standardized DSLs and System Design formalisms can interact? What level of similarity/differences they present? Can this interaction permit the recongifuration of AI carrying components in a system, how?**	**Can a UML system model and the relations between its parts be represented in a formal DSL ontology? How such a specification would look like? Can instances of the model be checked against these formal specifications for compliance? Can the reasons for non-compliance be established thanks to reasoning?**	(1) **Success=** Define a Domain specific and standardized ontology for CPSs.	Define a suitable ontology, for a subdomain of CPS. It is desireable that the ontology complies to some existing standard. Detail and assess its suitability and its usage with respect to the tooling environements.	We have developed **standardized domain specific ontology for UAVs, called** ODrone and integrted it into IEEE1872 (CORA) standard. **SUCCESS**
	How can we integrate formal knowledge into MBSE tooling envirnonements?	**Can a feasible technology stack be defined to implement this integration? Under which methodology this integration can be achieved?**	(2) **Success=** Define an integration methololgy specifying: how to use and integrate standardized ontologies and domain domain specific ontologies. Propose an implementation workflow.	Establish the limits, benefits and similarities between the system design formalisms (UML/SysML) and the domain specific ontology.	We have provided means to exploit this ontology (ODrone) via UML profiles within the Papyrus environement.
				Specify the mapping from the ontology constructs and their UML/SysML counterparts.	We have analyzed the constructs, diagrams and intended semantics of UML constructs and diagrams, as well as the concepts and relations in ODrone.
				Specify the technological resources neccessary for an implementatoin, and systematize their interaction as guidelines/methodology.	We have selected and justified the selection of UML Class Diagram and UML Composite Structure Diagram constructs as the sources for the transformation of the UML model into a OWL version, tha complies with ODrone's (and therefore CORA) semantics.
			(3) **Success=** Apply the methodology and define a feasible Technology Stack.	Define and justify a mapping from the ontology to UML, and from UML to the ontology	The mapping is not exhaustive and not unique. Not all the combinations of constructs have been targeted, but a sufficient mapping to cope with ODrone terminology has been provided. This can be enriched for the current use-case, or used as a template for other domains. Success, but room for improvement and experimentation.
				Define a workflow, necessary technological resources and clear inputs/putputs as well as stakehodelrs	A workflow using UML, OML and OWL with 5 main steps has been detailed. This worlflow enables the integration of DSL ontologies into UML, and the ewport of a UML annotated model

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

				necessary to implmenet the approach.	into OWL.
		How can the approach be tested/demonstrated?	(4) **Success =** Implement a Demo/ POC	Success= A proof of concept (POC) that shows the feasibility of the integration of formal DSLs into a MBSE tooling environement. A demonstration of the exportation of a W3C compliant version of the system model, and of some reaosning tasks on top of it.	As a POC we have specified a methodology, a technology stack and implemented a demo, along with its documentaiton, which permits:
				Test model designs for reconfiguration, and establish their compliance to the system specification, via reasoning.	a) The "import" of a formal DSL into the tooling environement, thanks to OML adapters. IN this case the ODrone ontolog.
					b) The specification of mappings from ODrone to CORA, and viceversa.
					c) Availability to use ODrone concepts and relations via a UML profile within Papyrus.
					d) The automatic creation of complex concept definitions from the UML diagramm, capturing the specifications of the system, and translated into OWL, while prserving the semantics of ODrone.
					e)Export of the UML model into OWL
					f) Examples of consitency and isntace checking of the exported model. Example of reuse of expert konwledge.
Safety and Hazard Analysis					
How to ensure safety of AI based systems that must made autonomous decisions in any operational situation	**How to define the scenario space in which the system must operate ?**		Success = Define the set of operating conditions (with their limits) under which the system is able to operate as intended		We propose a methodology and a tool support to define the operational design domaiin of an AI-based autonomous system. The methodology let define also the ODD limits trhough exclusion of the conditions (and combinations of such consitions) that are out of the system operational domain and by constraining the values of the conditions within the operational domain
	How to ensure that within its operational domain, the system will perform as intended	**How to prevent the system from any undisered behavior within its operational domain ?**	Success = Propose a hazard analysis approach to tackle the particular conditions/scenarios that may lead to adverse and undesired behavior from the system		We develop a hazard analysis approach that allow to identifiy any potential source of accident coming from classical HW/SW hazards, from any sensors limitations, from any model insufiiciencies, from any system's misuse, and any environmental events and safety variables and margins that the system must meet.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

	How to detect that the system will still perform a safe behavior when it is out of its operational domain	**How to detect that the system is out of its operational design domain**	Success= Monitor the ODD limits at runtime		Proprose a framework for dynamic risk assesment that : 1) includes a monitoring module capable to track relevant environments and system's events that were defined within the ODD and identify at design time as critical points for the system performance (safety variables) 2) Includes a safety enforcement module capable to fire safe decisions in front of unsafe or unknown scenarios, i.e. when the monitoring functions detect that the monitored variables are not anymore within the desired safety margins
		How to ensure that the system will take appropriate decisions to stay in a safe mode when outside of its operational domain	Sucess = define safety strategies and rules to give back ontrol to human, to navigate in a safe path, or to safely stop the system		
			Sucess = Implement a demo/POC		Develop a POC for hazard identification in AI based systems that is using the system'S ODD as input for the operational scenario generations
					Develop for runtime monitoring component that enables to enforce dependable decision at runtime. The runtime monitoring framework has been validated trhough simulation engines.
Verificatoion					
How can safety be formally guaranteed at design time?	**How can this process be modular enough to allow for tailoring the AI-based system to the needs of specific applications?**	**How can the lower-level implementation of a verified AI-based system keep the same level of guaratees?**	Success= Demonstrate feasability on a POC	Generate an AI-based system with high level of confidence in the satisfaction of safety constraints and target a low-level, embedded-software relevant implementation	We developed ColibriCS, a Constraint Library for Certified Solvers. Which allows to modularily define and prove separate parts of symbolic-AI artifacts that can be joined together and generate tailor-made AI-based components. Several domains have been added and the POC assesses promising prospects. A C low-level implementation can be automatically generated in order to best suite an embedded application.
How to bridge the gap betwee the various tools and cover a wider range of AI-based components (other than NN)	**How to integrate more than juste one method to offer a more diversified coverage of the validation process**	**How to encourage collaboration around one platform and allow extensibility**	Success = Integrate a wide range of tools in a one platform	Design and implement a modular, extensible, flexible, centralized open-source platform and integrate a wide range of tools	We developed CAISAR platform, for Characterizing AI Safety And Robustness. It was built on-top of ISAIEH (Inter-Standards AI Encoding Hub) which allowed it to have an expressive core that is able of describing more sophisticated properties that what was previously possible. CAISAR evolved beyond that and can now cover several AI-based technologies. In particular, it integrates propery-based testing (e.g. through AIMOS, the AI Metamorphism Observing Software), abstract interpretation based analysis (e.g. Through PyRAT, the Python Reachability Assessment Tool) and SMT based formal proof (e.g. through Marabou). CAISAR also handles properties on SVM (e.g. through SAVer).

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

					CAISAR has been released open-source.
Publications					
Dissemination and inter-partner interaction.			**(5) Success** = Articles, publications, inter-partners collaboration and dicussions.		Presentation of the article **"Knowledge Integration into Model Driven Engineering"**, in the OnUCAI workshop at the KR2021 conference.
					Current submission of article **"Augmenting Model-Based Systems Engineering with Knowledge"**, to the MDE Intelligence workshop, held at the Models 2022 conference.
					Accepted paper at EDCC 2022, **Using Operational Design Domain in Hazard Identification for Automated Systems**, Guillaume Ollier, Diana Razafindrabe, Morayo Adedjouma, Simos Gerasimou and Chokri Mraidha Accepted paper at DSD 2022, **Skeptical Dynamic Dependability Management for Automated Systems, Guillaume Ollier**, Fabio Arnez, Ansgar Rademacher, Adedjouma Morayo, Simos Gerasimou, Chokri Mraidha and François Terrier
					Accepted paper at ICRA 2022, **Towards an Uncertainty-Centric Dynamic Dependability Framework for Autonomous Systems,** Fabio Arnez, Guillaume Ollier, Huascar Espinoza and Ansgar Rademacher
					A cross-domain framework for Operational Design Domain specification, Guillaume Ollier, Morayo Adedjouma, Simos Gerasimou, Chokri Mraidha, ERTS 2022, Toulouse, June 2022
					A combined knowledge and simulation-based approach for identification and evaluation of unsafe scenarios for autonomous systems, Guillaume Ollier, Morayo Adedjouma, Simos Gerasimou and Chokri Mraidha, ICML 2021 (Poster)
					An ODD-based Hazard Identification for Artificial Intelligence based Cyber-Physical Systems, Guillaume Ollier VEHITS 2022, April 2022 (Tool Demo session)

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

					ReCIPH: Relational Coefficients for Input Partitioning Heuristic. Serge, Durand and Augustin, Lemesle and Zakaria, Chihani and Caterina, Urban and François, Terrier, ICML's WFVML 2022
					CAISAR: A platform for Characterizing Artificial Intelligence Safety and Robustness. Julien Girard-Satabin, Michele Alberti, François Bobot, Zakaria Chihani, Augustin Lemesle. IJCAI's AISafety 2022
					PARTICUL: Part Identification with Confidence measure using Unsupervised Learning. Romain Xu-Darme, Georges Quénot, Zakaria Chihani, Marie-Christine Rousset. (ICPR's XAIE 2022)

Table 1: CEA KPIs decomposition

Deliverable D5.6 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

2.1.3 Lessons learned

During the development of the project, there were some specific challenges to overcome, which only became known once the approaches and tools interaction were studied in detail. We present them next, to help overcome future research and implementations.

Ontologies and their integration: When tackling a specific domain (e.g. drone design), it is often not evident to determine which terminology and semantics should be selected. This effectively increases the effort to implement a KB based system design. Vocabularies and standards are not only diverse, but not all are available for free access, and only a few are available in the form of formal computer exploitable resources (e.g. OWL). It is important to verify the full scope of the target vocabularies, their level of formalization, their licenses and availability. We have provided an end-to-end solution for coupling DSL into standardized formal vocabularies, and their integration into MBSE tooling, effectively enabling the system designer to describe its system in terms of the ontology. Thanks to the bi-directional mappings, our approach also enables the export of the model as a W3C compliant representation. The approach and the implementation serve as guidelines on how to choose, integrate and exploit these resources.

Viewpoints and abstraction levels misalignment: Due to the complexity of cyber-physical systems, many stakeholders and viewpoints need to interact. It is important to define *a priori* the overlapping and interdependent information between the expected partners/tools to interact. Specially, the viewpoints and abstraction levels are challenging. For example, the physical components of a system influence on the functions a system can provide, which should satisfy the requirements. Nevertheless, concepts like *camera* or *sensor* can mean different things in different viewpoints (in the functional viewpoint it can refer to a camera driver, which is a portion of software, whereas in a physical component viewpoint it will refer to the actual device). Likewise, the requirements and properties of a system might be expressed at a higher level than the functions being handled by the designer. Thus, appropriate aggregation mechanisms have to be defined and justified to enable the desired interaction.

2.2 ANSYS

Cyber-Physical Systems (CPS) are a new generation of systems combining intensive connectivity, embedded computing and local intelligence, to create a link between the physical and digital worlds and allow cooperation between systems. As product complexity grows, so does the challenge of integrating individual components within a system to ensure they work together as expected.

2.2.1 Developments

During this CPS4EU project, Ansys focused on creating a complete digital prototype to understand and optimize the critical interactions between physics, controls, and the environment throughout the product development process. The chain of tools proposed by ANSYS was defined to extract from simulation data a surrogate model also called reduced order model. These tools form a machine learning toolbox which is physics agnostic. It can accept as input any data from any physics simulation solvers.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

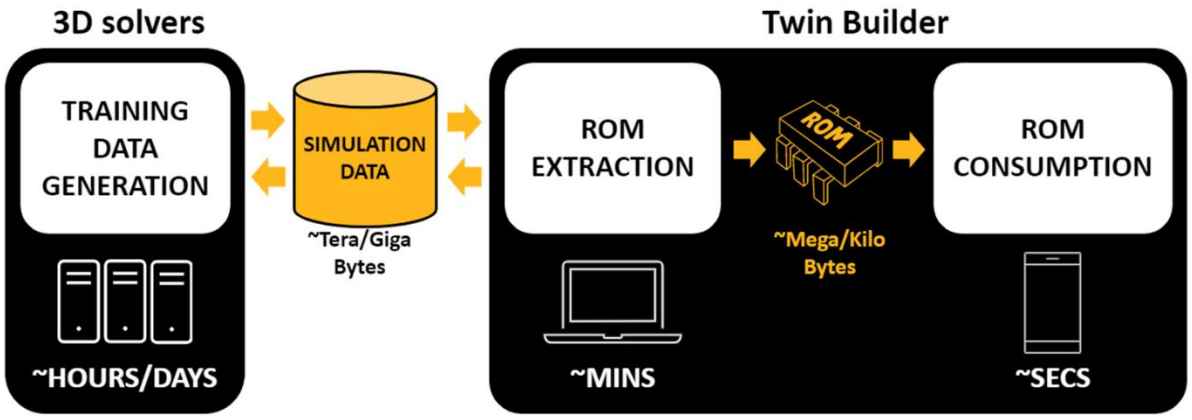


Figure 4: Machine Learning Workflow

Focus was made on integrating the three key tools for simulation with and adaptation of the execution support to the CPS target platforms selected by the partners:

1. Twin Builder:
   To build your system easily and quickly, Twin Builder combines the power of a multi-domain systems modeller with extensive 0D application-specific libraries, 3D physics solvers and reduced-order model (ROM) capabilities. When combined with embedded software development tools, Twin Builder allow you to reuse existing components and quickly create a systems model of your product.
   To validate your system and ensure expected performance, Twin Builder combines multi-domain systems simulation capabilities with rapid human-machine interface (HMI) prototyping, systems optimization and XiL validation tools.
   To connect your twin to test or real-time data, Twin Builder easily integrates with Industrial Internet of Things (IIoT) platforms and contains runtime deployment options, allowing you to perform predictive maintenance on your physical product. It is the only product that offers a packaged approach for your digital twin strategy.

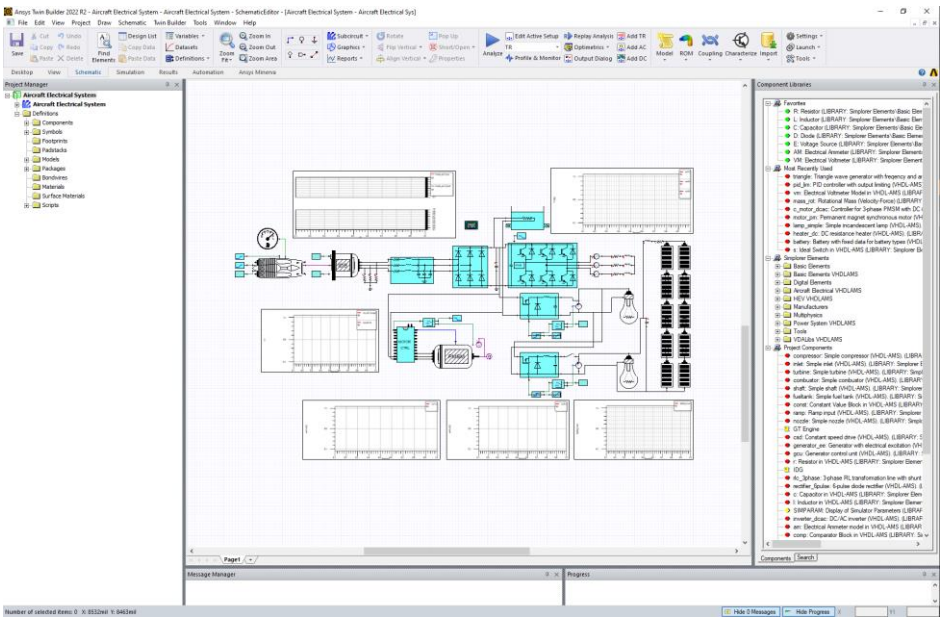


Figure 5: Ansys Twin Builder Schematic

2. Static and Dynamic ROM Builders:
   Twin Builder couples with Ansys physics-based simulation technology to bring the detail of 3D simulations, as reduced order models (ROMs), into the systems context to generate accurate and efficient system-level

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

models. Twin Builder uses ROMs produced from Ansys structural, fluids, electromagnetics, and semiconductor products to model mechanical assemblies; electromagnetic actuators and machines; circuit and cable parasitic; thermal networks; and signal integrity. ROMs can also be imported from a variety of third-party tools.

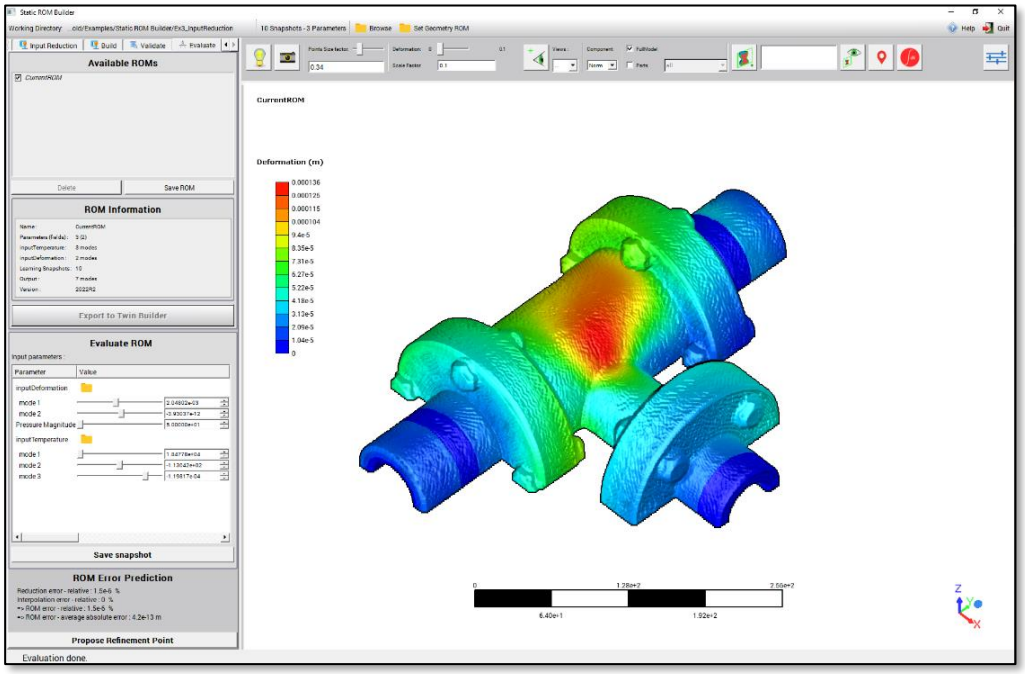


Figure 6: Ansys Static ROM Builder

This last tool is specifically designed to simplify twin validation by enabling the inspection of each of its composing parts, tune models parameters before deployment through on-the-fly simulation and cross-platform compilation to faster deploy the twin on specific real edge machines.

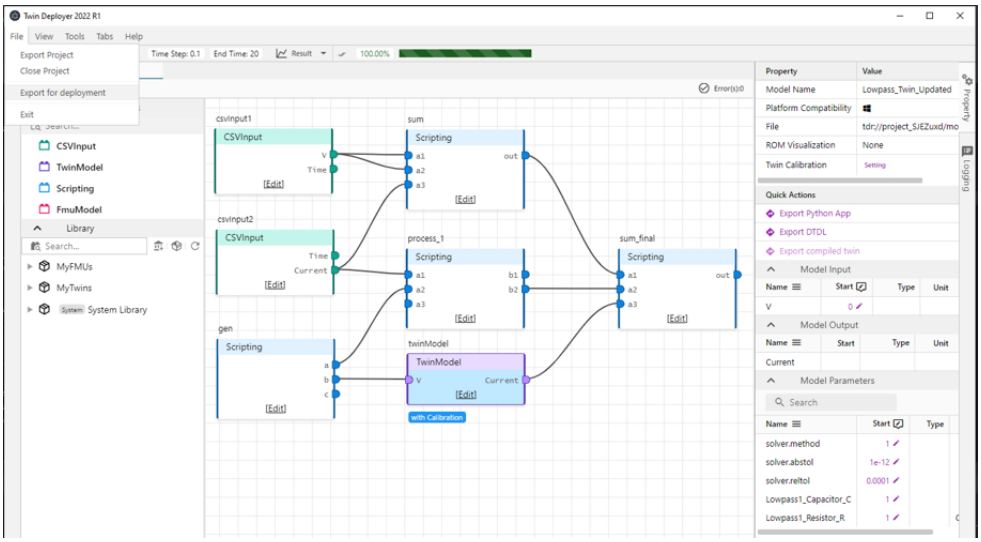


Figure 7: ANSYS Twin Deployer

Thus, this chain of 3 products takes as primary inputs results from physics simulation solvers and provide a guided workflow from these datasets to a model ready to be integrated into CPS platform. This set of tools is referred as the Twin Builder suite. First step of this chain begins with the use of Static ROM Builder or Dynamic ROM Builder that represents the ROM extraction tools. Static ROM Builder is specific to parameterized dataset. Dynamic ROM

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Builder is specialized in unsteady ROM extraction. From these specialized products, a ROM can be created and exported to be part of a wider system description as required by CPS vision. When this integration is finished and validated the whole model or part of this model can be exported to Twin Deployer to be prepared for an integration on any platform and OS. The generated object is self-standing and can be entirely embedded inside any project using Functional Mock-up Interface (FMI), which is standard that defines an interface to exchange dynamic models using a combination of XML files, binaries and C code zipped into a single file.

However, to be integrated inside CPS platform/framework, a federate provided by our partners shall be used via a Communication Federate Library to communicate with the High-Level Architecture (HLA), allowing them to publish and subscribe to the simulation data and synchronize in time.

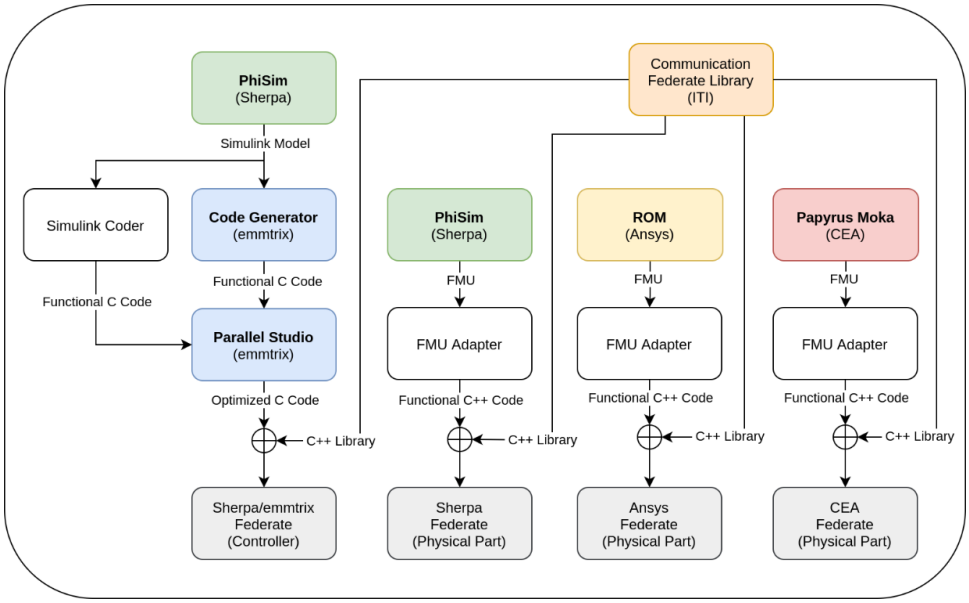


Figure 8: Simulation Components Generation

A new capability called Hybrid Analytics was also developed by Ansys but could not be integrated in the simulations from CPS, due to its late appearance in the CPS timeframe. This approach is providing calibration capability to tune simulation model parameters, so the simulation outputs match measured data.

2.2.2 Cluster

Ansys participated in the heterogeneous co-simulation only. This simulation is a hybrid vehicle system example proposed by Sherpa. Ansys is providing models obtained using data provided via 4D simulation using Ansys Software. These models are coming in form of co-simulation FMUs to add a deeper physical insight into the global model and thus by bringing a more complex and precise physical behaviour enhance the whole system predictions.

Inside an electrical motor, multiple interactions must be represented to account for reality. The state of the art is to account for them using simplifying assumptions. However, the precision obtained using those assumptions are not enough to consider real embedded applications. 3D simulations as to be considered mostly due to the non-globality of the coupling in between thermal effects and electromagnetic calculations (see Figure 12).

The focus here is on the electric power losses generated by the motor operating in a specific range of parameters and its coupling with the thermal heating and cooling of the electrical engine part of the use-case.

As an example of this usage, we provided a model of the Nissan Leaf Electrical engine. Ansys Provided 3 ROMs:

1. A transient ROM based on Ansys Maxwell calculation and providing a field view of the electrical losses. However, as this model is representing the variations of electromagnetic fields due to the engine rotation and the current variation, its usage timestep restricted the order of the milliseconds. Therefore, because the Sherpa use-case is made to run on a timeframe of hours, a second step was to be used: An averaging of the system response to cope with timestep in the order seconds.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

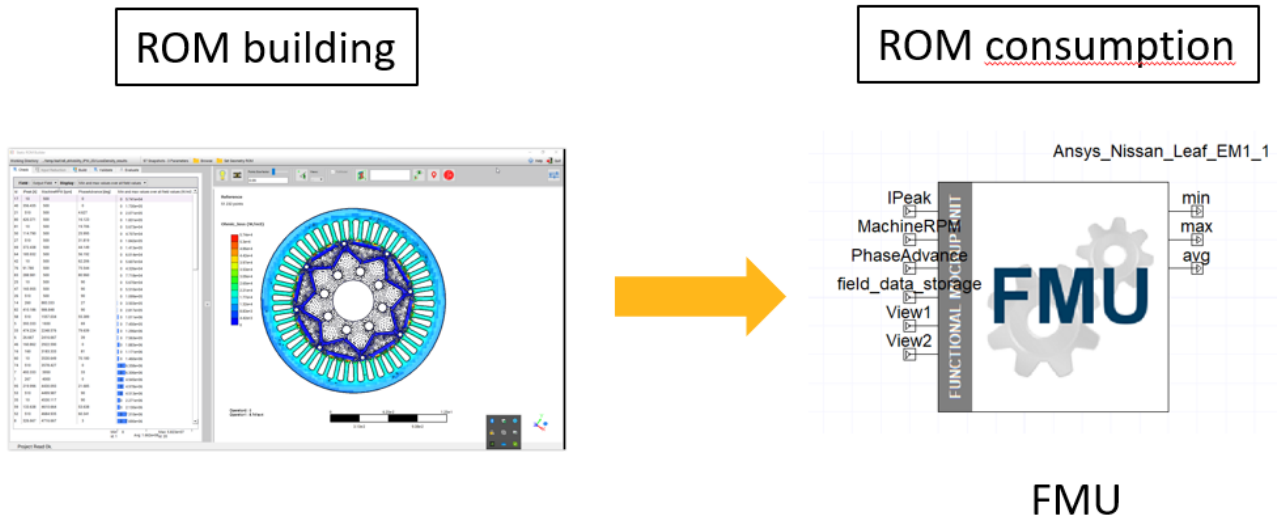


Figure 9: Dynamic ROM creation and co-simulation FMU generation

2. A parametric model enabling a faster control and calculation but giving a lesser insight on what happens in the engine.

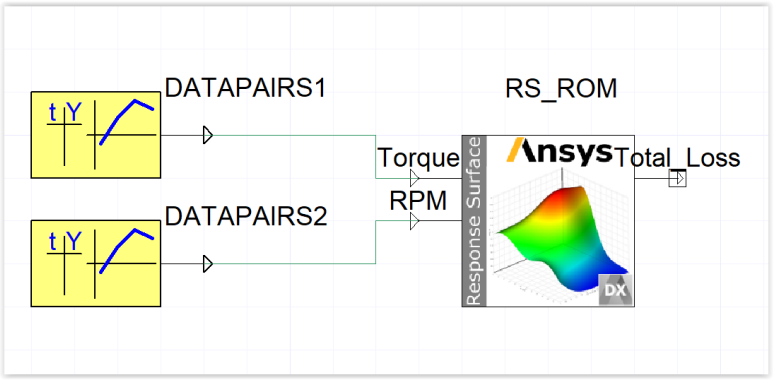


Figure 10: Resulting Response surface

3. A Thermal characterization of the electrical engine is done based on data extracted from Transient Ansys Mechanical calculation results and commanded by the electromagnetic losses which are set as thermal sources

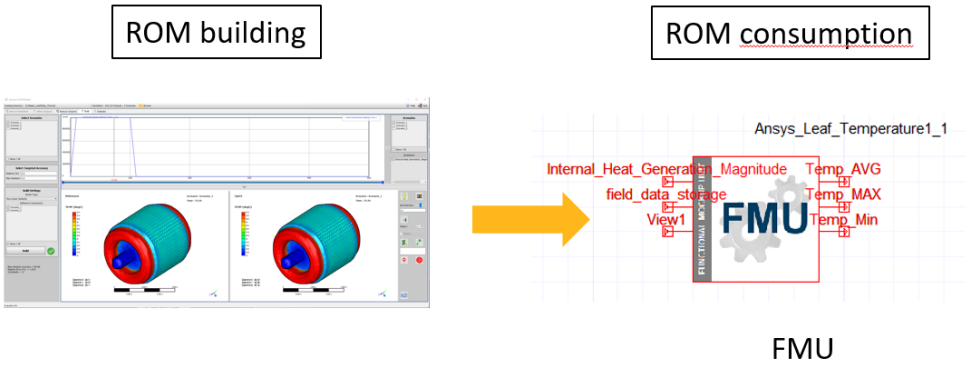


Figure 11: Dynamic ROM creation and co-simulation FMU generation

Regarding the specific results of the hybrid vehicle demonstrator, the general workflow dealing with physical ROMS is illustrated below:

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

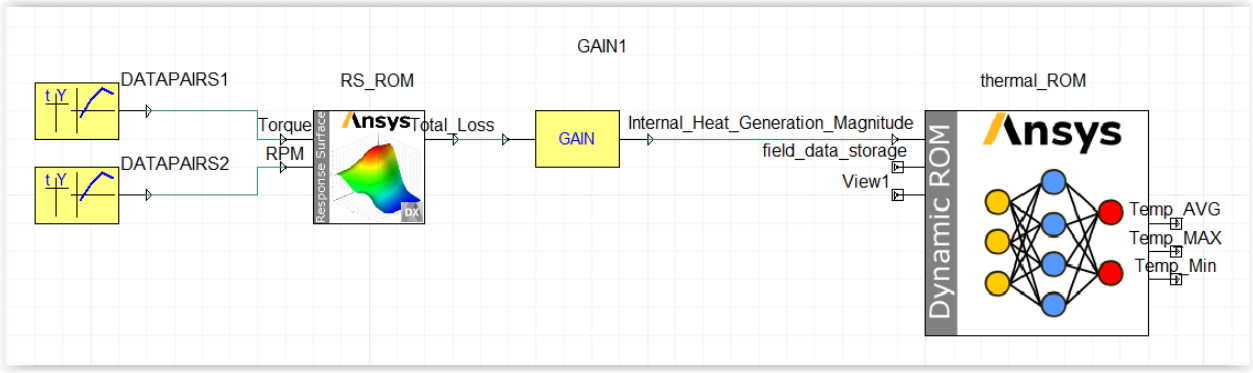


Figure 12: ROM workflow for thermal assessment of Leaf electric motor

2.2.3 KPI

Question	Proposed success criterion/criteria	Result
Development of non-intrusive and general reduced order modelling techniques to accurately approximate multiphysics non-linear and transient simulations by real-time applications	Success = (Accuracy better than 2% which means that using at least 3 use case CPS models we measure the relative error between reference results given by 3D physics solvers and real time applications; that relative error between the 2 methods must be lower than 2%) AND (Computation time between the 3D physics solvers and the ROM application divided by more than 1000)	On all the examples done in the frame of CPS, the precision of the model was under 1%. As most of the models runs in quasi real time, the saved calculation time is of course tremendous. The equivalent calculation without ROM was not done and could not be done in a reasonable timeframe. The evaluation of this last success criterion is just based on learning calculation comparisons (4h32 vs 7s: x8233) **Success**
Development of simulation based digital twins combining reduced order models and physical sensors measurements	Success = (Using at least 3 use case models accuracy better than 5% between simulation based digital twins and physical sensors measurements)	No measurement available for the test case at hand. Other experience done inside Ansys leads to an accuracy of 5% average with the presented ROMs and an accuracy of less than 2% if a hybrid ROM is used. Variations are accounted for depending on the non-linearity of the physics and the quality of the simulation and measurements. **Success to be confirmed**
How to achieve successful deployment of our simulation in CPS architecture	Success = Seamless integration of our FMU	FMUs were proved to be perfectly integrated inside our cluster. Modification, however, were done to ensure the compatibility with HLA architecture **Success**

Table 2: Ansys KPIs decomposition

2.2.4 Lessons Learned

CPS4EU was an extraordinary occasion to apply and develop the machine learning ideas we had in our research

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

group. The industrialization of Model Reduction tools is a challenge especially when field data is considered. We were given the opportunity to develop a ROM building, validation, and deployment suite as well as access to industrial partners and ideas. The development of FMU/FMI format were, up to now an obligation. We had the opportunity also during the project to discover the High-Level Architecture (HLA) proposed by partners and its quality in providing a common architecture for our simulation. The Run-time Infrastructure (RTI) that provides a standardized set of services through different programming languages with information exchanges, synchronization and federation management and the Federation Object Model (FOM) that specifies the Object Classes and Interaction Classes used to exchange data makes it a very promising tool to integrate our solvers and methods at Ansys.

2.3 INRIA

2.3.1 Introduction

During the CPS4EU project, INRIA directed its research to tackle challenges faced when designing, implementing, and deploying CPS. Our research advocates using verification techniques that operate at runtime. Techniques based on runtime information are complementary to the traditional techniques operating on source code or binary, i.e., static techniques. More particularly, we focus on the so-called runtime verification and runtime enforcement. While runtime verification allows checking some behavioural properties on the system, runtime enforcement allow enforcing certain forms of properties on the target systems by modifying their behaviour.

At the centre of our contributions to this project is THEMIS, our dedicated tool for decentralized verification. As CPS are in essence distributed and cooperating systems, THEMIS is a leading tool dedicated to monitoring that explicitly account for the decentralized and distributed nature of such systems. Various challenges arise when monitoring such systems such as finding the best way to organize monitors by considering variables such as the system architecture of the different components, the type, size, and number of messages required between monitors. Another important factor to consider is how to decentralize the specification. THEMIS facilitates designing and monitoring by allowing the user to design, implement, execute, and compare different monitoring algorithms. It can be used to monitor safety requirements on the connections between objects and the system and evaluate whether global properties hold on the system while not disrupting the existing architecture. Moreover, THEMIS and monitoring in general is an effective technique to provide complementary trust in the system by supplementing existing verification techniques that would not scale. Finally, THEMIS can also contribute to augmenting the safety of systems by providing some form of automation in the use of fail-safe fallback mechanisms on systems.

2.3.2 Work

2.3.2.1 THEMIS-BIP Integration

We worked to interface THEMIS with the BIP framework developed by UGA, namely DR-BIP, which is designed for dynamically reconfigurable systems. We now provide a unified flow for modelling and verification utilizing DR-BIP's modelling and THEMIS' verification approaches. We initially considered integrating with DR-BIP in two ways: online and offline. So far, we completed the offline integration layer with DR-BIP. The adaptation layer is summarized in the Figure 13. DR-BIP provides traces that capture the configuration and the states of components at the various execution steps. We created an adapter that parses the DR-BIP trace and generates compatible trace files for THEMIS. In this way, we can run THEMIS on these traces and monitor the system offline. We implemented the adapter as a Java library.

Below is an example run of the trace adapter developed on a use case of moving objects. The DR-BIP trace contains data about the state of each drone component. Based on the AP definition defined for a drone, the data is converted via the trace adapter into THEMIS trace files with AP definitions.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

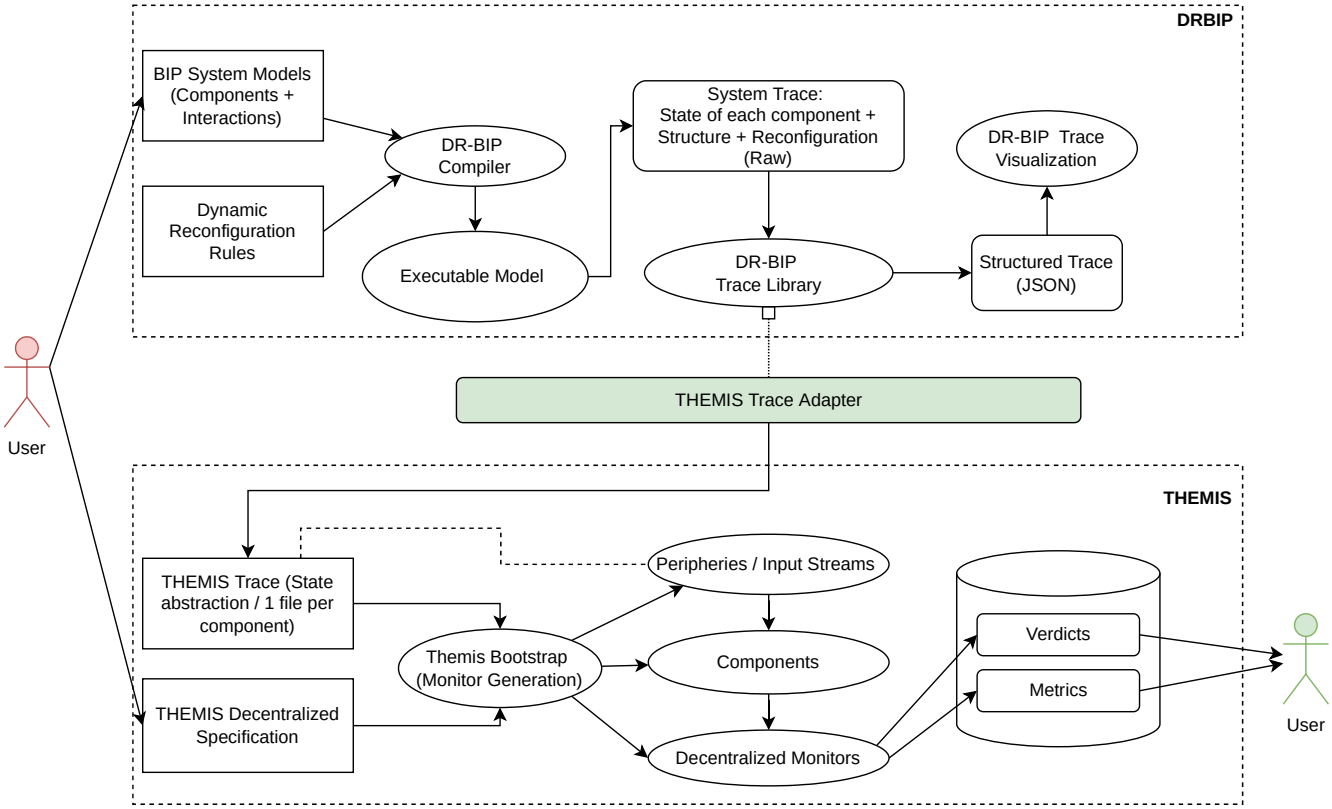


Figure 13: THEMIS-BIP TRACE ADAPTER

```
<TRACE> <DRBIP>  component DRONE_0_0 {
_m__pos:4.2,_m__v:0.2,_m__minSteps:246 }
<TRACE> <DRBIP>  component DRONE_0_1 {
_m__pos:3.9,_m__v:0.2,_m__minSteps:223 }
<TRACE> <DRBIP>  component DRONE_0_2 {
_m__pos:3.6,_m__v:0.2,_m__minSteps:220 }
<TRACE> <DRBIP>  component DRONE_0_3 {
_m__pos:3.3,_m__v:0.2,_m__minSteps:338 }
<TRACE> <DRBIP>  component DRONE_0_4 { _m__pos:3,_m__v:0.2,_m__minSteps:285
}
<TRACE> <DRBIP>  component DRONE_0_5 {
_m__pos:2.7,_m__v:0.2,_m__minSteps:274 }
<TRACE> <DRBIP>  component DRONE_0_6 {
_m__pos:2.4,_m__v:0.2,_m__minSteps:302 }
<TRACE> <DRBIP>  component DRONE_0_7 {
_m__pos:2.1,_m__v:0.2,_m__minSteps:310 }
<TRACE> <DRBIP>  component DRONE_0_8 {
_m__pos:1.8,_m__v:0.2,_m__minSteps:249 }
<TRACE> <DRBIP>  component DRONE_0_9 {
_m__pos:1.5,_m__v:0.2,_m__minSteps:230 }
<TRACE> <DRBIP>  component DRONE_0_10 {
_m__pos:1.2,_m__v:0.2,_m__minSteps:337 }
<TRACE> <DRBIP>  component DRONE_0_11 {
_m__pos:0.9,_m__v:0.2,_m__minSteps:206 }
<TRACE> <DRBIP>  component DRONE_0_12 {
_m__pos:0.6,_m__v:0.2,_m__minSteps:265 }
<TRACE> <DRBIP>  component DRONE_0_13 {
_m__pos:0.3,_m__v:0.2,_m__minSteps:244 }
```

Figure 14: DR-BIP Trace

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

```
DRONE {
    okstate: v >= 0.0 && pos >= 0.0
    stopped: v == 0.0
    slow   : v >= 0.0 && v <= 0.4
    fast   : v > 0.4
    fspec  : v < 0
}
```

Figure 15: THEMIS AP Definition

The generated trace files can now be used for monitoring with THEMIS.

```
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
okstate:true,stopped:false,fspec:false,fast:false,slow:true
```

Figure 16: Generated Trace File

2.3.2.2 Scenario-Based Simulation - Monitoring Gazebo Simulations

In the Scenario-Based Simulation cluster, which is directly related to the WIKA use case, we worked on integrating the SES and Gazebo tools from TUC, DR-BIP, and THEMIS. The integration aims to feed DR-BIP with specific scenarios created within SEStools. Starting from the generated scenarios, DR-BIP can be used to specify high-level model scenarios for a given protocol and output useful traces that can be monitored and verified by THEMIS.
We also worked on directly monitoring, with THEMIS, traces generated by Gazebo. To that end, we analysed traces generated from a Gazebo simulation that simulates a ROS node with obstacle avoidance where RPLidar is used for the detection of obstacle. The results below show monitoring a Gazebo simulation for drone obstacle collision. The Gazebo traces (column 1) are converted into Themis compatible traces, as per the APs (atomic propositions) definition below (column 2). Themis traces consist of a sequence of events (column 3) where each event consists of a set of atomic propositions. In this scenario, we are interested if the drone is within safe distance from the obstacles in the 3D space (house, tree, tower crane). For each obstacle, we calculate the distance between drone and the obstacle at each timestamp and check if it is below a certain threshold. In this scenario, the monitor registers to one component only that is the drone. We monitor to see if the drone is always in the ok state and get the verdict (column 4).

Gazebo Traces	Themis Drone APs Definition	Generated trace file for drone	Property verdict: G(okstate)

Sim Time : 3511.603000
house
Xaxis: -0.123079
Yaxis: 0.144468
Zaxis: 0.0
pine_tree
Xaxis: 5.45805
Yaxis: -9.63246
Zaxis: 0.0
tower_crane
Xaxis: 10.1279
Yaxis: 7.58691
Zaxis: 0.075655
iris_drone
Xaxis: -7.99991147992
	DRONE{
	moving: x != x_last || y != y_last || z != z_last
	collision_house : sqrt((x - house.x)^2 + (y - house.y)^2 + (z - house.z)^2) < threshold_house
	collision_tree : sqrt((x - tree.x)^2 + (y - tree.y)^2 + (z - tree.z)^2) < threshold _tree
	collision_tower: sqrt((x - tower.x)^2 + (y - tower.y)^2 + (z - tower.z)^2) < threshold _tower
	okstate: !(collision_house || collision_tree || collision_tower) && ! moving
	}
		moving: true, collision_house: false, collision_tree: false, collision_tower: false, okstate: true
		
		moving: true, collision_house: false, collision_tree: false, collision_tower: false, okstate: true
		
		...
			True
			
			True
			
			True

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Yaxis: 4.41162278954e-08 Zaxis: 0.0543868196363 Sim Time : 3511.610000 ...			

Table 3: Verdict table

Other properties can be monitored within this scenario; we started sketching monitoring some pre and post conditions for a crane lifting a load task. Pre-conditions such as drone is placed at position (0; 0; 0), two cranes are ready with an object attached to them, and drone should be ready to receive mission tasks. Post-conditions such as load is placed at correct position drone returns to position (0; 0; 0). For the future, we envision adding the enforcement of some properties at runtime. Some commands can be delayed if their execution would violate a required property. For example, a drone spots is moving object under the load, hence the cranes must delay dropping the load to the ground. Another example, some wind is detected, and cranes should delay any action unless it is critical

2.3.3 Challenges

One of the challenges we faced in this project is obtaining the traces that allow us to proceed in monitoring with THEMIS. Given that the traces are extracted from a running system, we had to wait so that all integration and development between partners is at an advanced stage so that we can acquire traces from running prototypes. Another challenge we faced is in the feasibility of formalizing specifications into LTL formula. For example, it is not straightforward to specify an optimization specification into an LTL formula. Another challenge is in the feasibility of instrumenting the target executions to retrieve sufficient information (traces) as some of these systems do not have an interface that allows inspecting and extracting the state of the execution.

2.3.4 KPI

Our success criteria relate to the project KPIs 2.1 and 2.2. Focusing on integration and speeding up CPS development time. We integrate and utilize THEMIS with other partners on the projects to provide monitoring support for existing tools or use cases. We measure our ability to provide monitoring and enforcement that speeds up CPS development by observing the monitoring features added and provided to other tools and scenarios.

Monitoring speeds up CPS design time by providing tools for verification and debugging. Since our monitors are generated automatically from a specification, we speed up development time and early bug detection. We measure improving debugging by the automatic analysis brought by our tools and by checking the coverage of our automatic approaches, and the amount of automatic verification code generated.

Question	Proposed success criterion	Result
Design time automation and improved requirement analysis: scenario modelling and early use case exploration with monitoring. Number of use cases.	**Success** = {2 uses cases identified with planned integration with other tools}	**Success** The integration between THEMIS and other tools has been used in 2 use-cases: one with BIP and one with Gazebo.
Integration of tools with other tools (automatically or manually)	**Success** = { traces integration between the 2 tools}	**Success** Integration of THEMIS has been achieved with 2 tools: DR-BIP and Gazebo.

Table 4: INRIA KPIs decomposition

2.3.5 Lessons Learned

CPS4EU allowed interacting and sharing ideas with partners. Moreover than our exposure to the work of the different teams in Europe contributing to the project, we integrated our work with UGA and TUC.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

As there is a lot of collective knowledge in the project contained within numerous documents shared between partners, we learned the importance of speed in finishing shared tasks with partners. The element of speed is essential in maintaining a collective awareness to efficiently finish a task. In slow-paced tasks, we noticed a need to revisit, reread and discuss concepts that was already presented and discussed previously by the partner. Avoiding delays and long breaks between such tasks can save a lot of time. Maybe one thing we lacked was the notion of **sprints** borrowed from agile software development.

2.4 SHERPA

Sherpa Engineering is a System and Control Engineering company that develop tool-based methodology for the design and evaluation of complex systems. Therefore, the Cyber-Physical Systems (CPS) are at the core of our activities of modelling, simulation, and control design.

The developments in the CPS4EU project concerned:

- Deployment and interoperability with the compilation of our models, the generation of FMUs and the creation of GUIs
- The basis of a co-simulation environment allowing us to run our models independently of commercial software
- Enrichment of our simulation models by introducing Safety and Cybersecurity elements
- MBD development of a control model: consensus on modelling rules for standardization and use of verification tool

In addition, Sherpa Engineering has contributed to two Tools Clusters: Heterogeneous co-simulation and Scenario-based simulation.

2.4.1 Developments

2.4.1.1 Deployment and interoperability of simulation models

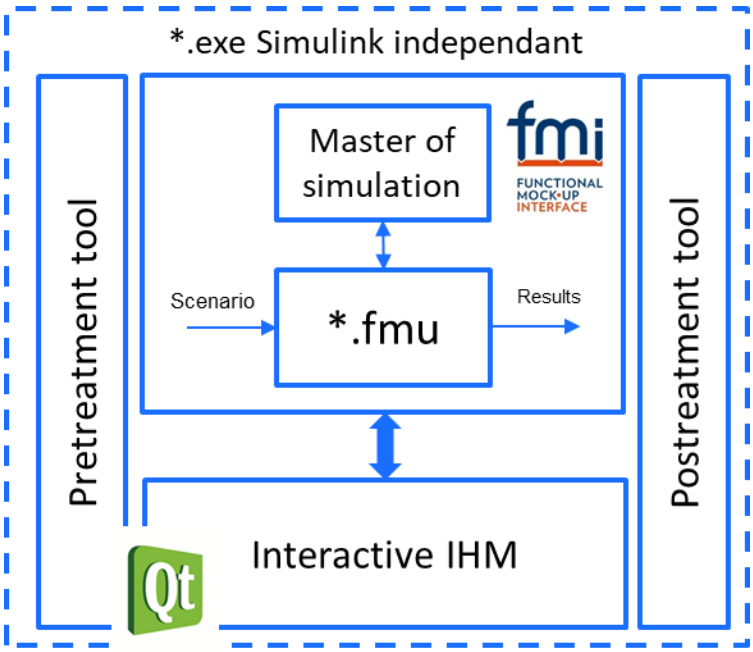


Figure 17: Integration and deployment scheme for digital simulators

The objective is to use the commercial modelling and simulation environment Simulink for the development of our simulation models and to deploy our simulators in standalone independently of Simulink.

As shown in the diagram in Figure 17, the simulator includes several components:

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

- Model of the physical system and the control system
- A real-time control GUI allowing to modify configurations and scenarios and to visualize the main observable variables of the system
- A pre-processing tool to define the scenarios
- A post-processing tool for results analysis and visualization

The implementation includes:

- The compilation of PhiSim models (multi-physics modelling technology developed by Sherpa Engineering)
  - The compilation is necessary for different uses: generation of FMU, use of Rapid Accelerator under Simulink, HIL validation
  - The main issues are: i/ management of S-functions developed in C/C++, ii/ Compiling and making available the source code and iii/ Management of the different compilers
- Deployment with a simulation platform based on a standalone executable
  - Two approaches were considered: i/ by using the Simulink Compiler and ii/ by generating FMUs and creating HMIs using an external tool (QT)

2.4.1.2 Co-simulation Environment

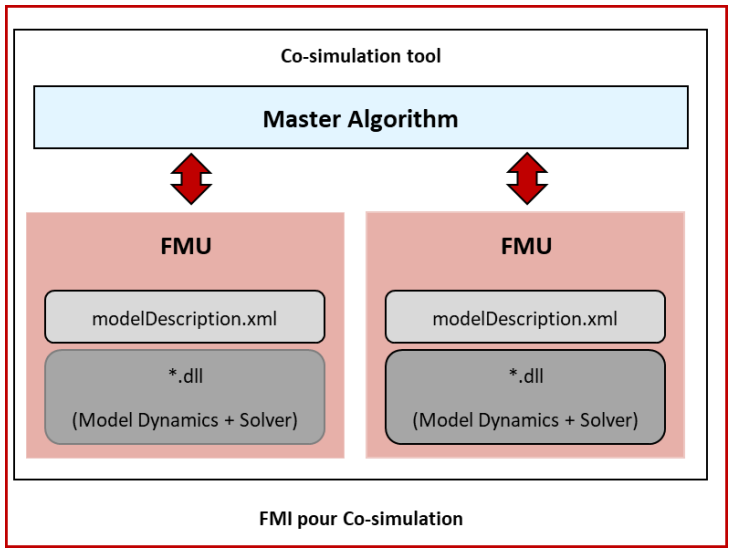


Figure 18: Co-simulation scheme using the FMI standard

The aim is to develop the basis of a co-simulation environment allowing us to run our simulation models independently of commercial software:

- The co-simulation environment developed uses the FMI standard which defines the interface of the models.
- The definition of the architecture is based on the SSP (System Structure and Parameterization) standard, which allows defining and executing a complex model with components provided by several modelling environments.
- Different use cases have been defined and used to validate the tool. These use cases represent generic examples that cover the vast majority of simulation models developed by Sherpa Engineering.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

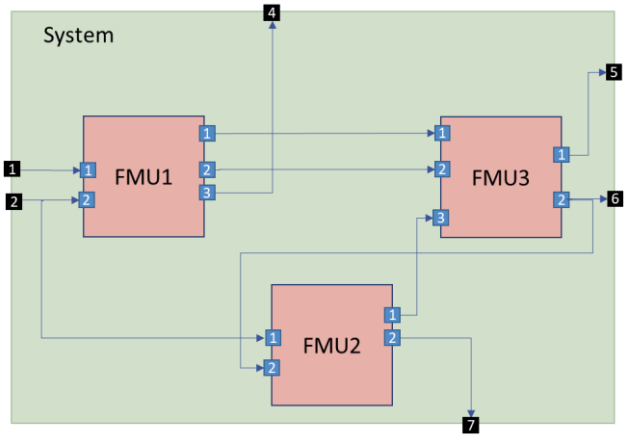


Figure 19: Example of an architecture with FMU components

2.4.1.3 Multi-facet modelling

The increase of multiplexed and connected systems (for the control part) as well as the integration of critical physical components, such as a fuel cell (for the operating part) leads us to consider Safety issues in our studies. The systems must improve their resilience to the risks of failure and to meet customer needs, our simulation models must be augmented with mechanisms allowing:

- introduce failures (breakdowns and attacks)
- dynamically simulate their effects (direct and induced) and monitor them (dashboard and scope)
- to try to detect these failures (identification of symptoms and setting up a diagnosis)
- to consider fallback modes against dysfunctional situations (degraded, refuge ...)

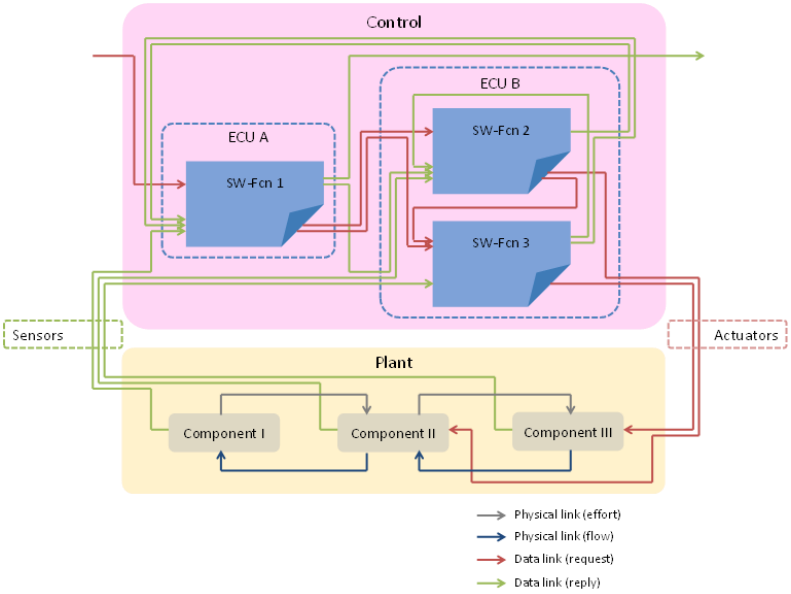


Figure 20: Controlled system architecture

In the context of the CPS4EU project, it is proposed to define these mechanisms according to a certain standardization. These elements concern the generic modelling of:

- the software functions (SW-Fcn)
- the generated error codes: Diagnostic Trouble Data (DTC)
- injection of erroneous signals (attacks)
- sensors and actuators (failures)

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

2.4.1.4 MBD design of a control model

Standards, such as ISO 26262 or DO 178 C, impose a development process to be applied from the first design phase of a product, thus involving the development of our models. The objective of these standards is to ensure the quality of the deliverables by applying a well-defined process at each stage.

Management plan for an MBD development project

The project management plan is a set of documents defined at the beginning of the project that lists the methods, tools, actions, and deliverables of the project team throughout the development phase. These documents cover at least the following topics:

- The quality assurance plan
- The development plan
- The validation and verification plan
- The configuration management plan

These documents are to be adapted according to the project (standard to be respected, additional customer constraint, etc.) and the SIL level to be treated and according to the equivalent document of the customer.

Tools

The use of tools for MBD development is essential to facilitate the work, but also to guarantee the development of the project. With some exceptions, these tools must be market tools that respect the project standard. These tools will be:

- Matlab / Simulink / Stateflow / etc. (Mathworks products).
- Targetlink / Control Desk / etc. (dSPACE products).
- MXAM (MES product).
- Or equivalent product (SpeedGoat, Vector, NI, etc.).
- Excel, SVN, Redmine, etc.

The use of these tools is not enough to comply with the standards, it is mandatory to follow the defined development process (Management Plan) and to customize the tools to the project. Sherpa Engineering has customized some tools on the market to better meet our needs:

- The Sherpa Engineering library allows to constrain the basic elements of Simulink with an initial parameterization adapted to the project.
- The data dictionary allows to define in an Excel file all the signals, parameters and important constants.
- Modelling rules allow developers to improve the quality of models based on best practices and rules.
- The customization of the development environment makes the activity easier for the developers.

2.4.2 Contribution to Tools Clusters

Sherpa Engineering has contributed to two Tools Clusters (*Heterogeneous co-simulation* and *Scenario-based simulation*) by providing a generic use-case and models.

Use case provided by Sherpa Engineering

The industrial objective is to design and evaluate the Energy Management System (EMS) of a Plugin Hybrid Electric Vehicle (PHEV).

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

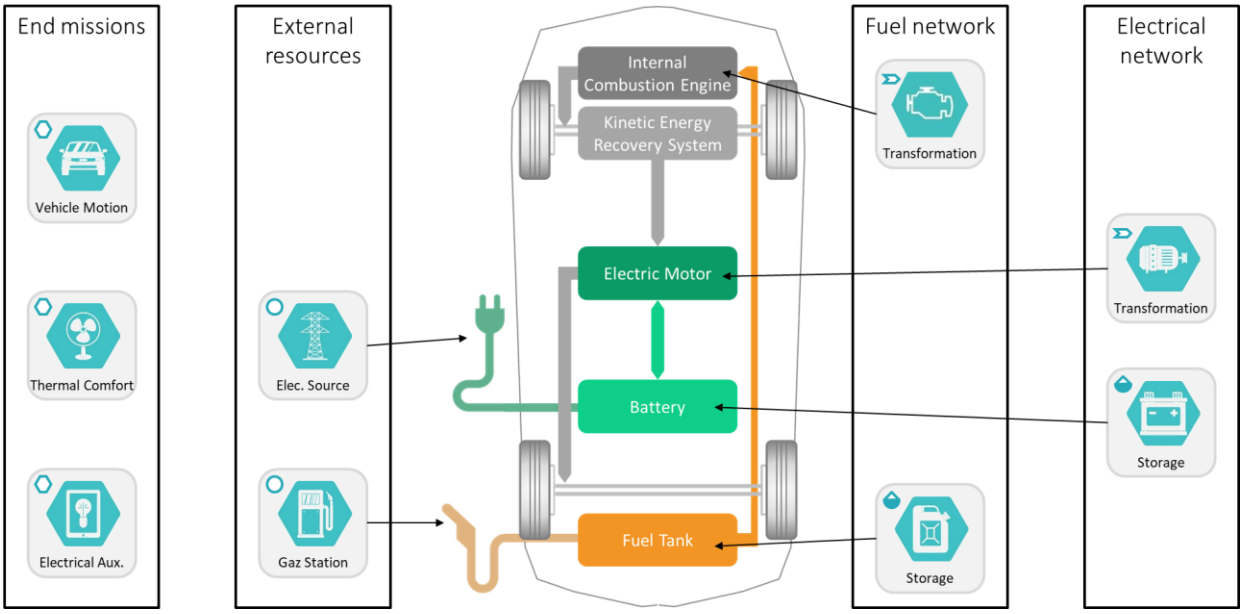


Figure 21: Energy Management System of a Hybrid Vehicle and its environment

Hybrid vehicles require an energy management strategy, which governs the drivetrain components. The objective is to minimize the fuel consumption subject to constraints on the components, vehicle performance and driver comfort. The energy management strategy plays a very important role in the improvement of fuel economy and the reduction of emissions. For a PHEV, requirements are mobility and comfort. These requirements are followed by three generic end-missions:

- Vehicle Motion: vehicle dynamics calculation in order to have a power need for making the vehicle move,
- Electrical auxiliary: vehicles every other load, power need is calculated as a power consumer,
- Thermal comfort: passenger thermal comfort in the vehicle, power need is calculated with outside and inside temperature.

These end-missions are controlled by a decision manager and a global resources manager.

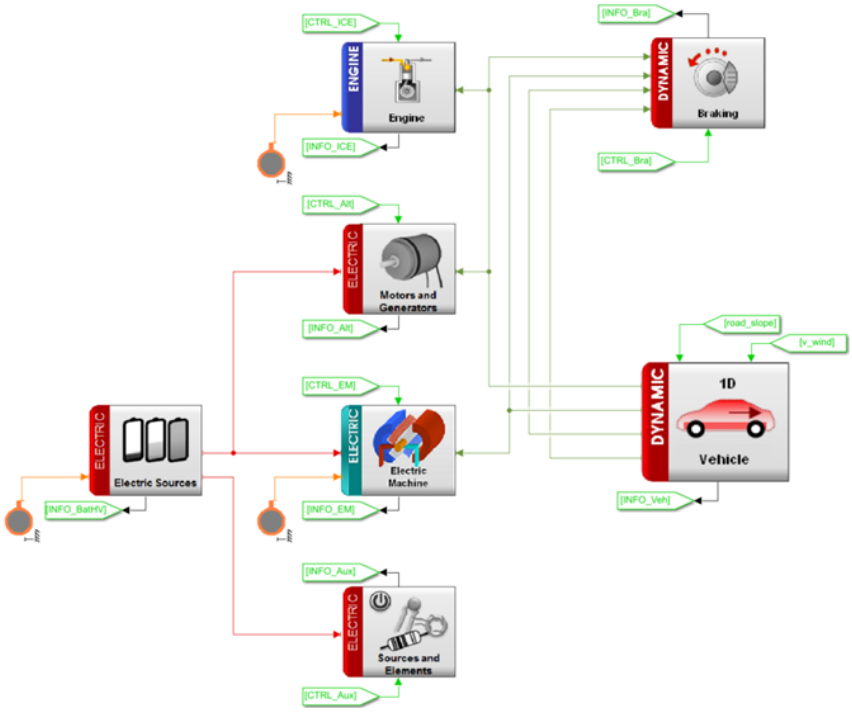


Figure 22: Electromechanical system design in PhiSim

A simulation model using PhiSim was developed. This model include the main physical parts of the PHEV:

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Component	Description
Engine	Gives the thermal motor shaft torque as a function of the engine angular velocity and the engine actuator position.
Brake	Represents the braking system. This element is a simple link between brake pressure and brake torque.
Motors and generators	This electrical generator is used for battery charging. It is an energetic macroscopic representation (torque and power losses are given in a table) of electrical machine.
Electrical sources	This advanced battery model considers Thevenin model for its electrical dynamic behaviour.
Electrical machine	This Electrical motor is used for vehicle motion. It is an energetic macroscopic representation (torque and power losses are given in a table) of electrical machine.
Vehicle dynamics	Model of longitudinal vehicle dynamics (include aerodynamic force, slope, but not tire slip).
Sources and elements	Represents the electrical auxiliaries (lights, windshield wiper,…)

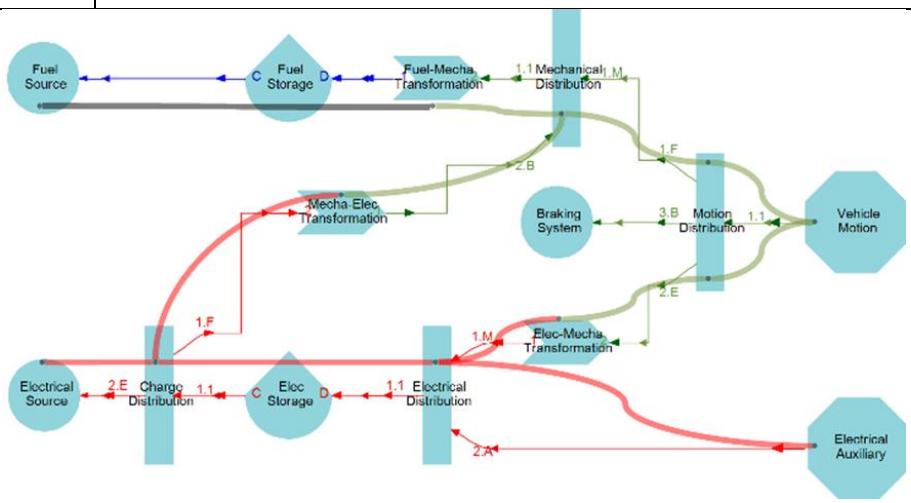


Figure 23: Electromechanical control strategy design in PhiSim

The model allows controlling the energy flows of a hybrid vehicle (mechanical - green highlighting - and electrical - red highlighting -). The distribution blocks (rectangles) allow the necessary energy to be allocated to consumers on the sources (electrical and fuel) according to a predefined priority. They also make it possible to distribute the energy produced to consumers (in this case mobility and electrical auxiliaries) according to a predefined strategy. These distribution blocks may contain either heuristic logic or a more advanced optimization algorithm.

The vehicle needs mechanical power requested by the vehicle motion. The mechanical power is provided either by transforming the fuel energy using the thermal motor (the green highlighting), or by transforming the electrical energy using the electrical machine (the red highlighting).

The battery (Electrical storage) is charged either from external electrical source or by transformation of mechanical power to electrical power using the electrical machine.

Contribution in the *Heterogeneous co-simulation* cluster

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

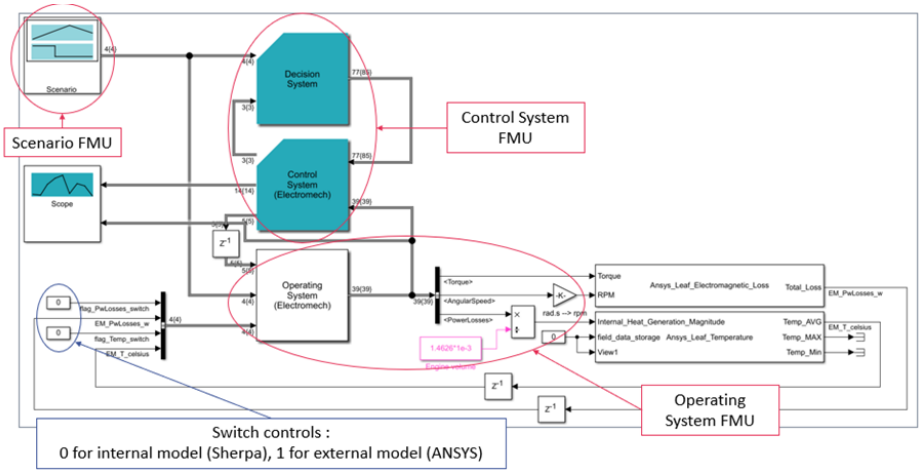
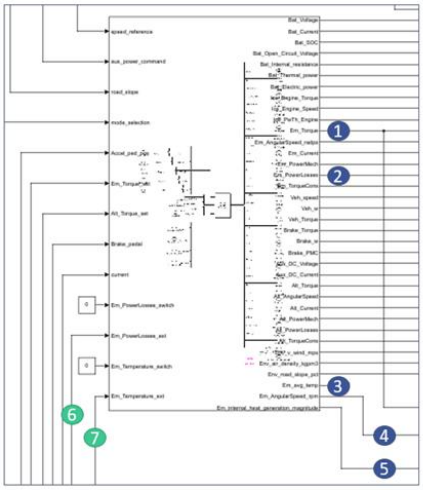


Figure 24: Final integration of ANSYS FMUs in Sherpa PHEV model

The objective was to integrate (or to link) a detailed 3D thermal model of the electrical machine provided by ANSYS to the system model developed in PhiSim. The technical objective is to improve the energetic evaluation by taking into account the evolution of the machine efficiency in respect to its temperature.

The workflow is as follow:

- First integrate the FMU coming from ANSYS with PHEV white box model,
- Second generate the Sherpa FMUs from PHEV model,
- Finally integrate the Sherpa FMUs with ANSYS FMUs in PhiSim and validate the obtained results before FMUs delivery.



- 1 : Electric engine torque (N.m).
- 2 : Electric engine power losses (W).
- 3 : Electric engine temperature (°C).
- 4 : Electric engine angular speed in rpm.
- 5 : Electric engine internal heat generation magnitude (W/m3).
- 6 : External electric engine power losses (W).
- 7 : External electric engine temperature (°C).

Figure 25: The FMU generated from the operational part of the PHEV

There are three FMUs from the PHEV model: scenario FMU, Control part FMU and the Operational part FMU.

The PHEV model is separated into three different models. These models are then prepared to the generation (adding the sources/sinks, configuration of the simulation parameters, etc.). The generation is done using Simulink coder R2021b. We required this version (or later) in order to be able to generated open code inside the FMUs.

The most important FMU is that representing the operational part of the PHEV. The generated FMU is presented in Figure 25; the ports concerned by the interface with the ANSYS FMUs are highlighted and illustrated.

The final step before the delivery of the generated FMUs is the validation of the integration results. For this we need to link the PHEV generated FMUs with ANSYS FMUs, launch the simulation with the appropriate parameters and verify the results.

Contribution in the *Scenario-based simulation* cluster

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

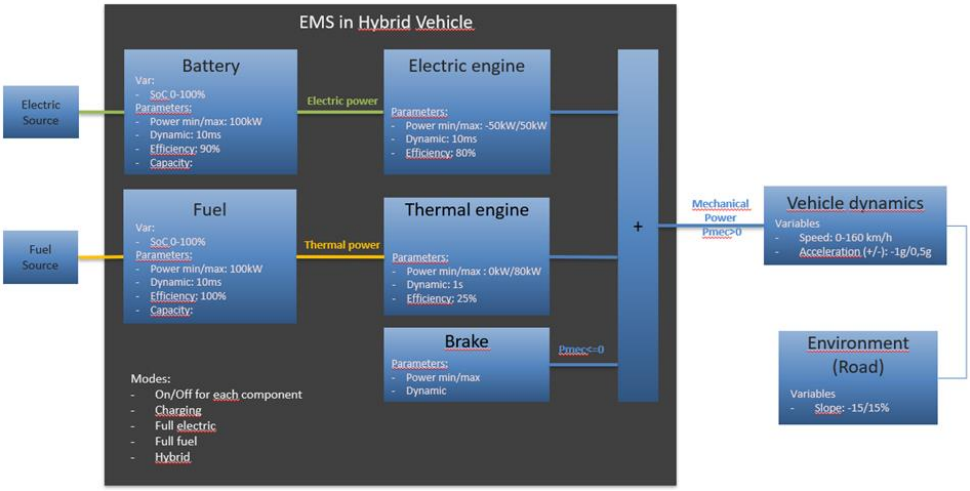


Figure 26: The energy management system in its environment

As a first step of the scenario definition, we opted for the specification of the EMS independently of the Hybrid Vehicle model. This choice is motivated by the fact that the extraction of the requested inputs from the whole PHEV model necessitates a lot of effort. The EMS is in fact specified in different parts of the PHEV model.

The EMS has three operating modes: Full Thermal, Full Electric and Hybrid mode. The variables which are taken into account to move from one mode to another are the speed of the Vehicle, the state of charge of the battery, and the mechanical power coming from braking/acceleration.

The EMS objective is to combine Energy optimization with Vehicle power efficiency. We need to check, in a given circumstance, if the EMS is in the right operating mode, and if the battery is used (discharged/charged) correctly by testing several scenarios.

For this first step, an EMS model was specified using Simulink Stateflow. Before the generation of the scenario from the SES tool, we validated the three scenarios defined using Simulink signal builder.

2.4.3 KPI

Question	Proposed success criterion/criteria	Result
Use of PhiSuite (from Sherpa) in an integrated way with techno providers' tools	Success = {validate the tool chain on an industrial use case (for instance RTE) using PhiSuite (Sherpa tools) with other techno providers' tools}	PhiSuite was used in different tasks and in interaction with several partners. First, PhiSim was used for the realization of a demonstrator for the case study of RTE (defined in their deliverable, WP9) for the optimization of the electricity network at the scale of a geographical area with production of renewable energies and storage. A simulation model, including an energy control strategy, compliant with RTE requirements was designed, evaluated and validated. Second, PhiSystem was used for system definition of hybrid vehicle (PHEV) at functional level and for the definition of the simulation architecture. The resulted model is used as a support by partners to understand the PHEV use case, as well as, as a starting point for the definition of the possible interactions between partners around the use case.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

		Then, PhiSim is used in: • The heterogeneous co-simulation cluster by providing the simulation model of the PHEV and the generation of FMUs. • The scenario-based simulation cluster for the design of the Energy management system (EMS) and the evaluation of scenarios given as input to the model. **Success**
Consolidate the Sherpa modelling and simulation workflow	Success = {detailed specification of a M&S workflow with which we will be able to provide a M&S service (what to do, how and with which tools) for a modeling and/or simulation request at any stage of the system engineering process}	Sherpa worked on the consolidation of its internal M&S workflow. A detailed specification of the M&S workflow was provided in the deliverable D5.2. This workflow is used partially during the project: • At the solicitation level: a system definition model was provided for the PHEV use case but no concrete application was done for the filtering of the system specification for a specific simulation request. • At the level of Composite model design: for the definition of simulation architecture in PhiSystem is done, the design of simulation models in different partners tools, then their integration in a final simulation model. • At the level of system Analysis: for the evaluation of partners scenarios and their validation against the expected results. To conclude, the workflow is validated partially on different needs. In order to validate the completeness/applicability/efficiency of the whole M&S workflow we need to apply it from the beginning to the end using the same use case. **Success to be confirmed**

Table 5: Sherpa KPIs decomposition

2.4.4 Lessons Learned

CPS4EU was an excellent opportunity to interface our methodology with complementary technologies developed by European partners. Several technical advances have been achieved and should be pursued to increase the maturity of these developments. The first advance is the connection between the system modelling generally used by Sherpa and a more detailed modelling provided by ANSYS. This connection allows considering the evolution of a subsystem subjected to excitations calculated by the global model. It allows introducing a magnifying glass effect in a relatively abstract model. We had also the opportunity to discover the High-Level Architecture (HLA) proposed by ITI. This model-interfacing standard has some interesting advantages over the FMI standard that we use more regularly. The use-case that we have implemented in the project has allowed us to remove the various obstacles and to adapt our development processes to make the use of HLA more fluid. Finally, we were able to test the scenario generation methodology proposed by TUC. This methodology is based on interesting semantic fundamentals and allows improving the consistency of the validation process of complex systems.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

2.5 EUROTECH

The EUROTECH tool suite has been adapted and evaluated in relation to the Leonardo use case; the results are reported in the WP8 deliverable.

2.6 TUC

A scenario describes the initial conditions and timeline of significant events. The elements of scenarios in a simulator include the systems and subsystems of interest (entities), the environmental conditions & the course of events

Scenarios are an essential part of the whole simulation engineering process. Due to the complex nature of Cyber-Physical Systems (CPS), simulation-based verification involves using scenarios as a cost-effective method. Scenario-based testing is already being used extensively in automated vehicles, especially validating Automated Driving Systems (ADS). Tools and standards such as OpenScenario, OpenDrive and OpenCRG illustrate the effort in this direction. Aviation has started using scenarios in the last few years, and a few working groups are developing a standard scenario definition language for aviation. The scenario definition language (SDL) used by TUC is one such example.

TUC is involved in exploring scenario-based approaches for simulation-based verification of different CPS. The aim was to leverage the tools and techniques developed by TUC across multiple disciplines.

2.6.1 Scenario-based approach

Throughout the project, TUC employed the approach of scenario modelling for various partners. The whole process starts with an operational scenario and ends with the execution of the simulation, where operational scenarios are descriptions of the operation of a system in plain language. These are translated into a structure capturing all the elements and their relationships to the scenario. TUC's SDL uses an ontology to capture these elements. The approach bridges the gap between people and systems. It can be used as a starting point for further development as a domain expands or the ontology embraces new or additional concepts. TUC's SDL uses a meta-model called System Entity Structures (SES).

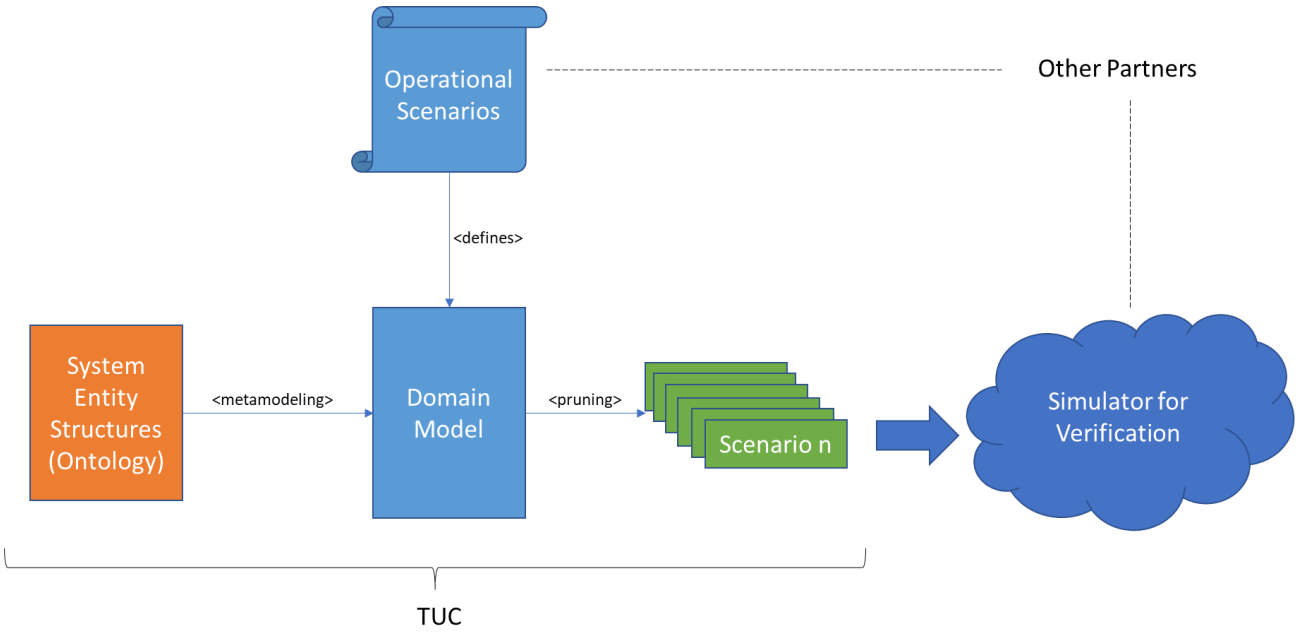


Figure 27 - TUC's Scenario Modelling approach

The main idea is to model all the elements of a scenario and their relationships using SES into a domain model. Once all possible scenarios are modelled in the SES domain model, selecting a particular scenario is defined by choosing a specific configuration through pruning. This resultant structure is in the form of a decision-free tree called a Pruned Entity Structure (PES). A PES represents a scenario in XML format that can configure a simulation

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

and assess the system for safety. A single domain model in SES can generate multiple scenarios. Parsing the scenario file leads to the configuration of the simulator. Hence, the scenario file is the test case executed on the simulation environment.

2.6.2 Tool Evolution

During the early phases of the project, TUC's SDL tool consisted of the SES editor and the PES editor to handle the different aspects of the scenario development process around domain modelling (SES) and scenario modelling (PES). The tool is now refactored and improved with new features and renamed the **Operational Domain Modelling Environment (ODME).** Operation Domain Modelling Environment (ODME) is a GUI Java tool that uses the system entity structure and a high-level ontology framework targeted to model, simulate, and design scenarios. There are significant improvements to the tool in dealing with functionalities of domain and scenario modelling and improvements in performance and usability. There are new sub-modes that target some extensions in the form of **Scenario management and Operational Design Domain (ODD**).

Domain Modelling mode represents knowledge of decomposition, taxonomy and coupling of system. It has a set of axioms and elements: Entity, Aspect, Specialization and Multiple-Aspect. An Entity is an object of interest (real or artificial component) and can also have variables attached to it. An Aspect denotes the decomposition relationship of an Entity node. Specialization nodes represent the taxonomy of an entity. A Multi-Aspect is a special aspect representing a multiplicity relationship that specifies the parent entity as a composition of multiple entities of the same type. Specialization and Multi-Aspect are represented by one, two and three vertical lines, respectively.

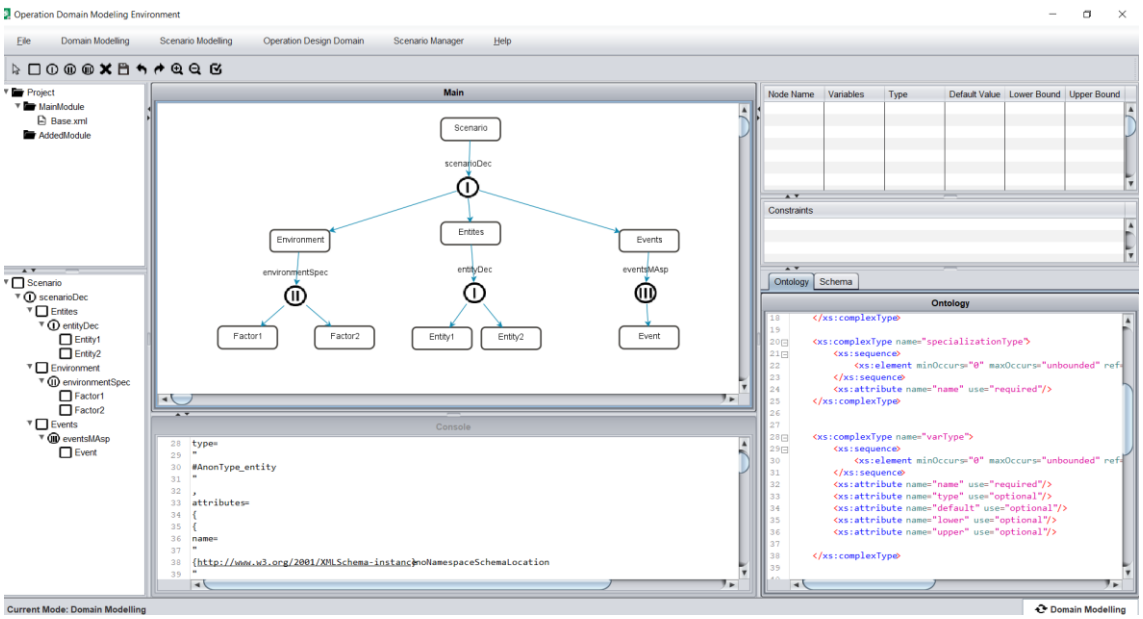


Figure 28 - Operational Domain Modelling environment

The figure above shows a simple domain model. It shows a simple *decomposition* (|) relationship where a scenario consists of Environment, Entities and Events, whereas the Entities consist of Entity1 and Entity2. The *specialization* (|||) relationship is shown in environment where it can have Factor1 or Factor2. Events show *multi-aspect* (||||) relationship where there could be multiple events under this section.

Scenario Modelling mode prunes the created domain model and generates different scenarios. Pruning is the operation in which a unique system structure is derived from a Domain Model, and the result is called Pruned Entity Structure. A Domain Model represents a family of models for a given application domain in terms of decomposition, component taxonomies and coupling specifications. In the domain modelling process, all the available options of a system are considered. As a domain model describes several system configurations, the domain model tree needs to be pruned to get a particular configuration. Pruning cuts off unnecessary structures from a domain model tree based on the specification of a realistic frame to bring this configuration, which is a selection-free tree. The pruning

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

process normally reduces a domain model by removing choices for an entity with multiple aspects and specializations consisting of multiple entities. A domain model tree can be pruned by assigning values to the variables, selecting one entity from various options of specialization node, and specifying cardinality in a multi-Aspect node. The figure below shows an example of pruning Multi-aspects. Similar features are available for other options.

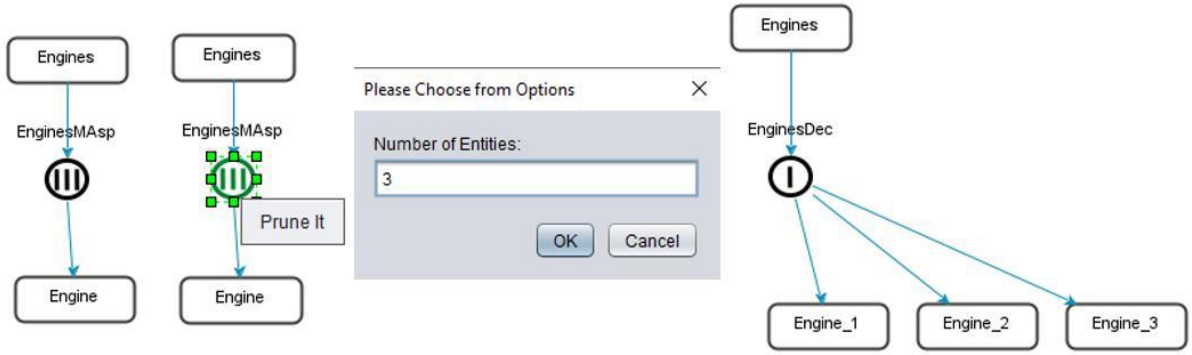


Figure 29 - Pruning Process

Scenario management includes keeping track of all the scenarios generated by one domain model. The users can assign classification metrics to each scenario. The user will also be able to generate scripts seamlessly to execute a scenario on the associated simulator like Gazebo, Anylogic etc. Valuable feature in progress is the automatic selection of scenarios based on certain characteristics equivalent to automated pruning. This is currently a hot topic for research in Scenario-based validation. The **Operational Design Domain (ODD)** functionality can extract the ODD for a particular domain model in form of human readable tables. ODD is necessary for providing operational context while defining scenarios, especially in AI-based systems. The format of ODD is currently under debate in the research community, but a human-readable table is a first step in this direction.

2.6.3 Cluster Involvement

The Scenario-based approach by TUC was the heart of the Scenario Simulation Cluster. The following figure denotes the interaction of the Scenario Definition Language (SDL) and its associated toolset with the other partners in more detail. Starting from the bottom, The SDL is used primarily for modelling Hybrid Electric **Vehicle events** in the case of Sherpa. The focus is on **environmental constraints** for the AirSim Drone Path Planning in the case of CEA. While modelling the factory floor, the aim was to demonstrate using a scenario model for various **entities** (machines, sheets and processes). For UGA and INRIA, the Scenario modelling tool was used indirectly through UC2 “Collaborative lifting” in Work Package 8. TUC’s SDL modelled the various scenarios for the interactions between the crane and the drone. UGA’s DR-BIP was used to model checking the protocol between the crane and the drone. INRIA provided the monitoring capabilities for the traces generated by the simulation.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

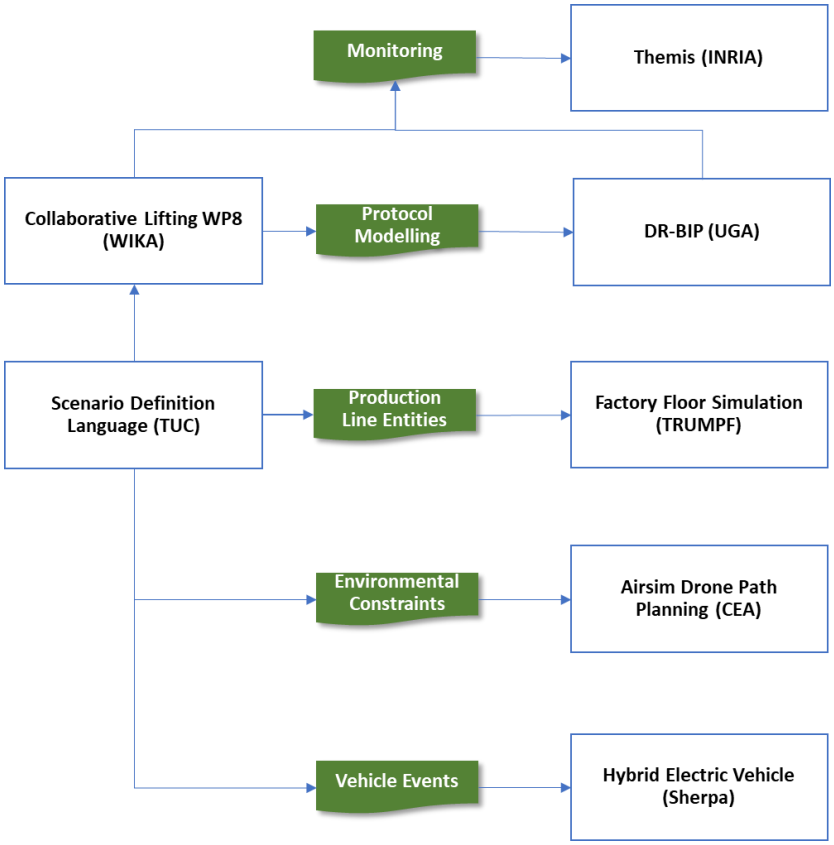


Figure 30 - Scenario based Simulation Cluster Partner Interactions

2.6.4 KPI evaluation

Question	Proposed success criterion/criteria	Result
Improved requirements analysis through an easier process of designing a scenario based on a graphical selection of available elements of the simulation and sharing of the scenarios among the stakeholders	Success = {all possible scenario elements for the use case modelled} AND {GUI tool to traverse through the possible elements to develop a particular scenario} AND {common interoperable format of the defined scenario such as XML}	The ODME tool allows for scenario modelling of a CPS's operations using a graphical interface. It can also generate an XML output for each scenario **Success**
Automating the process of visualizing the scenario for easier simulation-based verification	Success = {launch the elements in the simulator automatically using scenario file with no elements missing} AND {inject events at the appropriate timeline}	The XML scenario files were parsed to extract the configuration scripts for the Simulator. The scripts contained the setup elements for the simulator as well the necessary events to be executed. **Success**
Model all three aspects of scenarios(entities, environment and events) through different use cases	Success = {model entities, events and environment as a part of one-use case} OR {model entities, events and environment separately in multiple use case}	The different aspects were modelled in three different CPS' from WP5 partners , as well as all elements combined in the WP8 "Collaborative Lifting" use case **Success**

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Integration of open scenario infrastructure with multiple platforms	Success = {model all the elements of the simulator from at least two platforms	The Scenario infrastructure was demonstrated on four different platforms **Success**

2.6.5 Learnings and Future Work

The scenario-based approach using a common language was an important first step towards verifying different types of CPS. Modelling scenarios for varied environments and entities of CPS were a bit challenging initially, and we were able to demonstrate scenario modelling for each use case.

- The three elements of scenario modelling (entities, events and environment) do not apply to every use case. For example, there are no environmental factors for a factory floor simulation (TRUMPF), and the events are more like processes that are strongly tied to entities. For other cases like the HEV, the events for the car testing were the main focus of the testing
- Through scenario modelling of different CPS, there is now a basis for continued exploration for more complex systems
- The tool was enhanced to support domain modelling, scenario modelling and the management of scenarios. The scenario manager will be a crucial component in the scenario workflow.

Future work

AI-based systems have a related concept of Operational Design Domain (ODD) embedded into scenario-based testing. ODD defines the operating conditions the system/ subsystem is intended to perform. This needs further exploration.

Another area for improvement is the automatic selection of scenarios from the domain model using a specific classification criterion. This would significantly enhance the efficiency of Scenario testing.

2.7 TRUMPF

TRUMPF is involved in multiple work packages in the CPS4EU project. The tools developed in WP3 and WP5 are standalone products as well as they are applied in the demonstrator in WP8.

2.7.1 Developments

TRUMPF's main contribution to WP5 inside the CPS4EU project are the Simulation Model Library and the Simulation Configurator.

2.7.1.1 Simulation Model Library

The Simulation Model Library consists of material flow simulation models for a broad range of TRUMPF machine tools, automation units, and intralogistics components like storage systems and Automated Guided Vehicles (AGVs). The Simulation Model Library is implemented in the agent-based discrete event simulation environment Anylogic. Discrete event simulation means that the state of the system is not calculated after defined time steps but after each event that changes the systems state. In agent-based modelling, the system is not modelled from a top-level view but from the perspective of the agents, e.g. a machine or worker in the system. Each agent has its own behavioural model and is able to interact with the agents in his environment. This approach has proven to be very efficient in terms of computing performance and simple modelling of the interactions between the components in a complex production system.

Screenshots of the workflow how a simulation model of a production system is set up can be found in Figure 31. At first, the component is added to the model via Drag and Drop. The input and output ports of the agent are connected to the corresponding automation units or manual transport blocks. Afterwards the agent's parameters are set and a 3D visualization is added. To reduce the complexity for automatic simulation model generation that is part of WP8, we encapsulated the machine tool building blocks and automation components to preconfigured production cells. These preconfigured cells reduce the effort for model creation even further.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

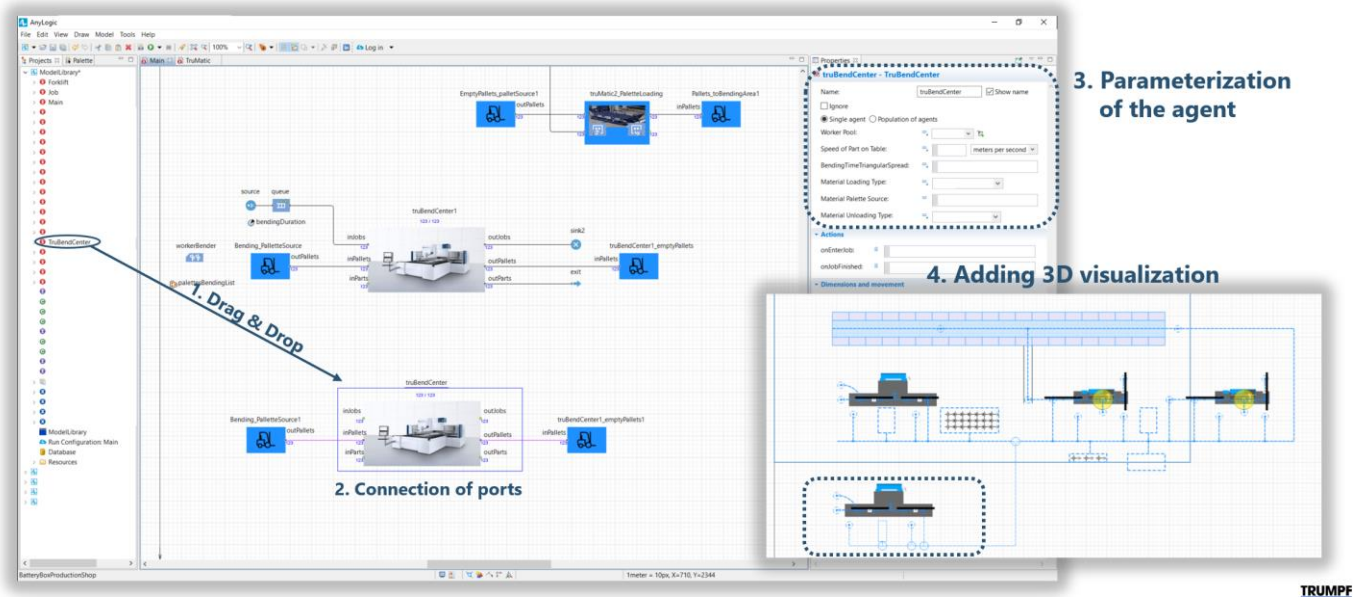


Figure 31: Workflow for creation of a simulation model from the Model Library

Moreover, an intralogistics controller is implemented. It uses the same material flow rules as the real Manufacturing Execution System (MES). It distributes work orders to the right machines and orchestrates multi-step production processes. This task is particularly complex in the sheet metal industry as every work piece can have a different material flow through the factory, the size of parts can vary from a few centimetres up to multiple meters and very small lot sizes are daily business of our customers.

2.7.1.2 Simulation Configurator

The second part of a simulation scenario is the production program that should be produced by the system. For each part, the process times and the process graphs must be defined; this is done using the simulation configurator tool. Moreover, the most import machine performance parameters can be set in this tool. Screenshots of the workflow are displayed in Figure 32. At first, an overview of the machines and their performance parameters is shown. In the second step, the process graph for each product is defined. Finally, the quantities and lot sizes as well as the start dates and due dates are defined. This information is transferred to the simulation via an xml document. The before mentioned material flow controller distributes the tasks among the available machines.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

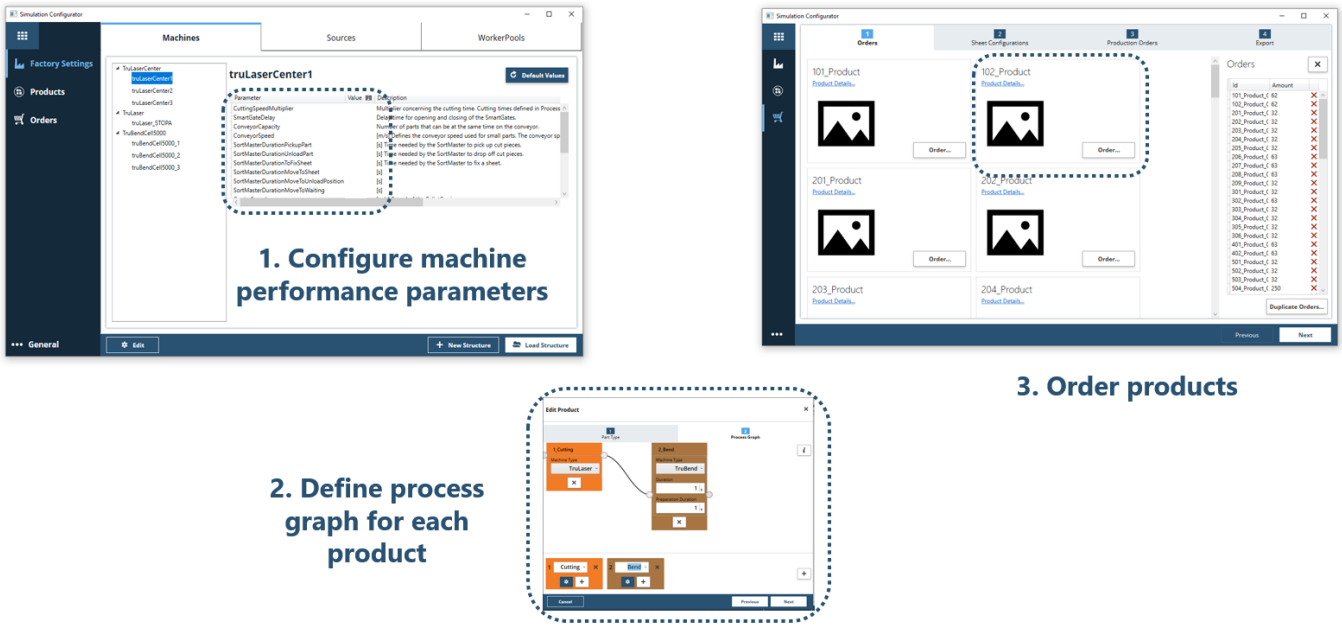


Figure 32: Workflow for creation of a production program using the simulation configurator

The results of the simulation are key performance indicators (KPIs) that are directly visualized in the simulation tool. Deciding indicators can be the throughput, the utilization of the machines and waiting times for material supply. An example KPI dashboard can found in Figure 33.

Figure 33: KPI Dashboard for a Machine

On top of that, the material flows are visualized in 3D. An example can be found in Figure 34. The visualization helps potential customers to understand how the material flows in his new factory will look like. The visualization enhances the trust in the simulation results.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

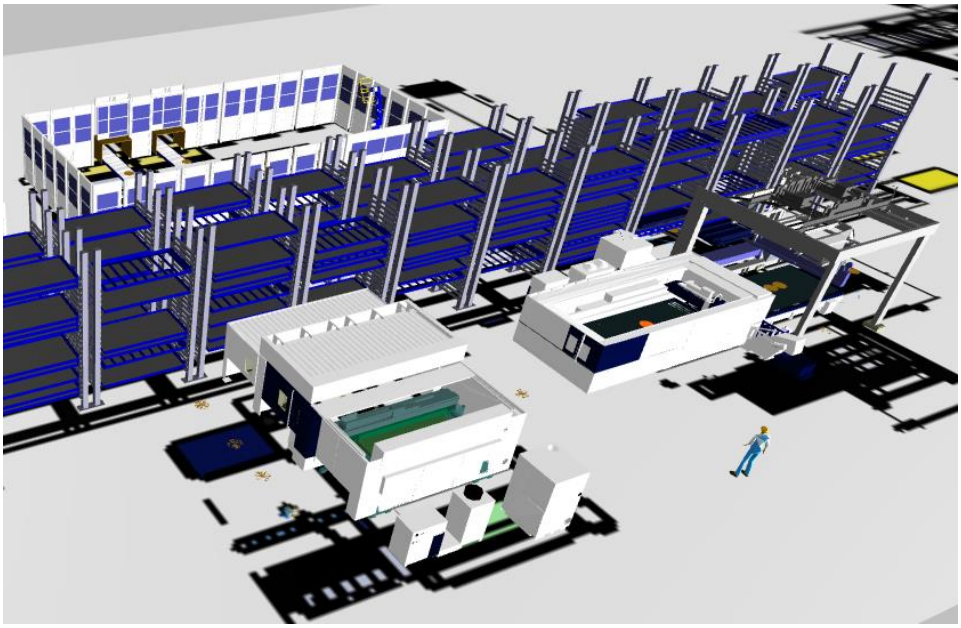


Figure 34: 3D Visualization of a sheet metal production system

2.7.1 KPI

Question	Proposed success criterion/criteria	Result
Is it possible to set up a material flow simulation with reasonable efforts?	Success = A standard simulation project can be completed within five workdays.	Various standard simulation projects have been completed with the aimed effort. Nevertheless, if customer request includes custom solutions the effort usually exceed five workdays. **Success**
Are the simulation results accurate enough?	Success = Each simulation building block provides a correct logic and the simulated times have a maximum deviation of 5 %.	All simulation building blocks have been validated regarding the correct logical behaviour together with machine experts, automation experts and control software expert. The validation regarding the correct lead time is still ongoing. **Success to be confirmed**
Successful application of the developed simulation model library in customer projects	Success = Application of the simulation model in at least 5 customer projects. The benefit of the simulation is proven by answering questions that could have not been answered in the offering phase.	The simulation model library was used in multiple customer projects in order to provide answers to complex material flow issues. **Success**

Table 6: TRUMPF KPIs decomposition

2.7.2 Lessons learned

CPS4EU enabled TRUMPF to support customer consulting by simulation. Customer queries that could only be roughly answered by analytical calculations can now be answered far more precisely. The development of the simulation configurator made it far easier for us to set up different simulation scenarios. This reduces the investment risks for our customers who are often small and medium enterprises tremendously.

Factory planning for our customers is just one initial use case. The developed model library has laid the basis for exploiting further improvement potentials. In WP8, we showed the first steps towards a real digital factory twin by automatically creating the simulation model from a 3D hall scan and using real time data. In the long term, this will

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

enable simulation based real-time optimization of production schedules that promises savings in building space, energy and sheet metal.

2.8 UNA

CPS are systems in which information and software technology are connected to mechanical components involving real-time data transfer and exchange as well as control or command via infrastructure such as the Internet. As a result, the safety requirements for cps are increasing as the likelihood of people coming to harm increases because they may be involved in the systems' processes. In addition, the attack surface increases as critical communication is carried out using public networks.

2.8.1 Developments

During the CPS4EU project, the University of Augsburg (UnA) developed a chain of multiple safety/security analysis and modelling options combined in one CPS analysis tool called MoCoAnalyzer.

2.8.1.1 MoCoAnalyzer

The MoCoAnalyser consists of three major components: Model-based analyses, code-based analyses and a modelling editor. Both categories of analysis require a holistic CPS as input, which is modelled using the modelling editor. As mentioned, the analysis is done in two different ways summarized in the following Figure 35:

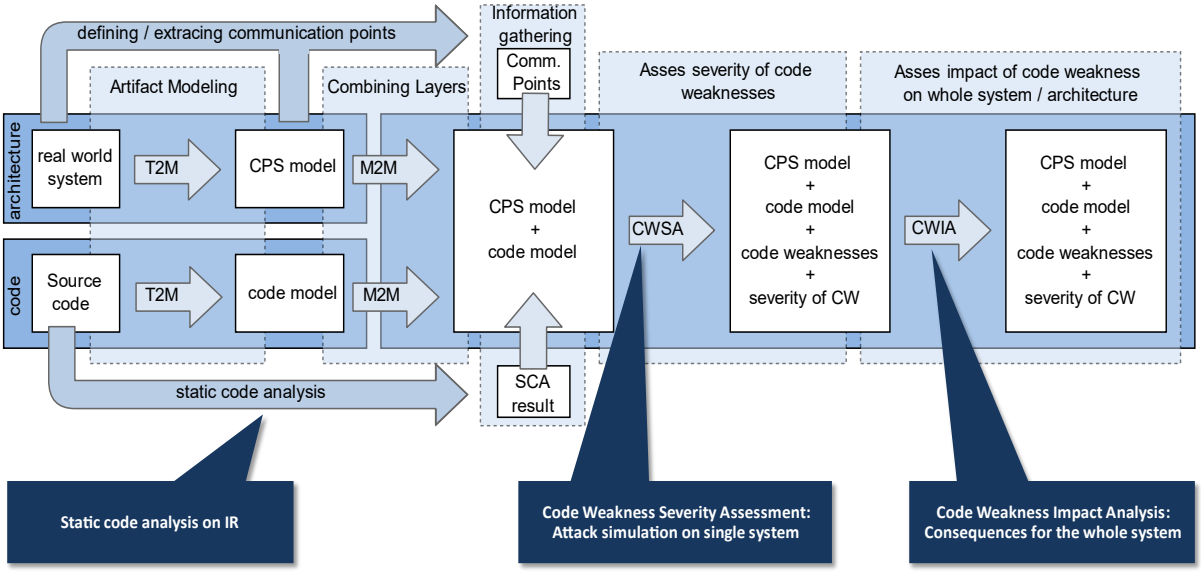


Figure 35: Combined model- and code-based analyses

On the one side, the model-based analysis starts with the real-world system, and through a Text2Model transformation, e.g., through architecture mining, or by hand modelling with the model editor, the CPS (architecture) model is achieved. This model allows defining or extracting the communication points.

On the other side, the source code of a system is transformed through a T2M transformation into a code model, taking the original e.g., C/C++ code into consideration but also the intermediate representation (IR) of the translated code. On this IR a first static code analysis is performed.

In the next steps both models, namely the CPS model and the code model, are combined to obtain a holistic view of the system, taking the architecture as well as the program code into consideration. Both first analyses (communication points and static code analysis) are integrated into the obtained model. On the combined model further analyses are performed, namely a code weakness severity analysis coming up with an attack simulation on a single system. Resulting in an annotated and integrated CPS and code model with code weaknesses and their severity. Afterwards a code weakness impact analysis is performed. Moreover, as described in detail on the model-based analysis further architectural analyses can be performed.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

2.8.1.2 Model Editor

To enable the deployment of UnA's Tools, the usage of the CPS meta model is required. The CPS meta model was developed by UnA during WP1 activities and was designed to be able to model the different CPS4EU use cases. The meta model is presented in detail in D1.2. A productive application of the meta model to map a use case model is given through a modelling editor. This modelling editor was developed during WP5 activities.

The main purpose of UnA's CPS Modelling Editor is to model and analyse cyber-physical systems. During the modelling process, the CPS meta model is used as a basic set of rules that are utilized to ensure the consistency of the model.

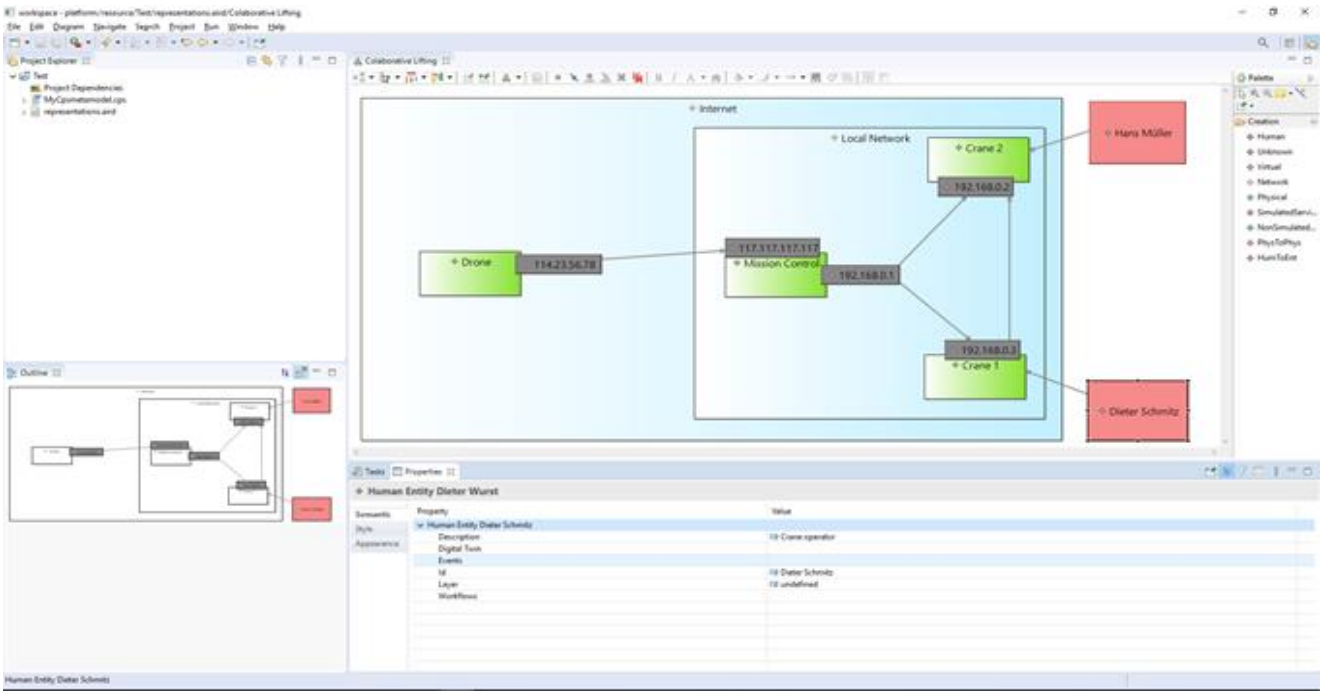


Figure 36: CPS Model Editor

The modelling of CPS is carried out by creating instance models of the CPS meta model. Therefore, the editing tool (Figure 36) provides some tools to instantiate the main aspects of a CPS on an architectural level, i. e. connections with the physical world, digital twins. The editing tool is based on Sirius as part of the well-known Eclipse Modelling Framework.

The created instance models represent the basis for UnA's analyses. The analyses are carried out on the instance models, considering that most analyses have different requirements on the model and therefore the model has to be enriched with needed information.

2.8.1.3 Code-based Analysis

The code-based analyses rely on CPS models interconnected with code models. The MoCoAnalyser supports the automatic derivation of code models by invoking the LLVM framework. The framework is used to compile and optimize input data into LLVM-related compilation artefacts. These artefacts are abstractly linked and lossless transformed into a code model.

To identify vulnerabilities in a CPS model, the code model must be linked to the CPS model. This is achieved by associating functions of the code model to services or machine-related entities of the CPS model. In more detail, a system can be roughly defined as a set of entities where each entity must interact with at least one other member of that set. Any interaction between entities follows some sort of protocol. In the case of human-to-human

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

interaction, natural language is usually used following a specific syntax and protocol. Machine-to-machine interaction is defined by protocols like IPv6. Machine-to-human interaction is also dominated by protocols normally described by manuals. As the code model contains the behaviour of one or more components of the system represented by the cps model, we focus on machine-to-machine interactions. The ability for system components to communicate with each other is usually provided by precompiled libraries. Therefore, functions that transfer data from or to such libraries are marked and manually associated with services and machine entities of the cps model. Based on this information, connections are derived that connect the components of the system on the code layer.

To detect code weaknesses and assess resulting vulnerabilities, UnA has developed three code-based analyses:

1. Static Code Analysis on LLVM Intermediate Representation (SCA on IR)

The first step is to identify code weaknesses. In contrast to vulnerabilities in cps models, vulnerabilities in code models follow patterns that are not specific to system architecture. Instead, several vulnerabilities of different systems can follow the same pattern. Such patterns are called weaknesses. They help to identify the core of a vulnerability whether the code snippet is exploitable or not. One of the most prominent sources of such patterns is the Common Weakness Enumeration. It is a community maintaining a list of software and hardware weakness type, this list is used by many providers of tools for static code analysis.

The Static Code Analysis on IR is a static code analysis that focuses on detecting code weaknesses on compilation artefacts. In contrast to traditional approaches, UnA focuses on analysis that is as close to the hardware as possible as compiler optimizations can induce weaknesses. Additionally, the analysis uses advanced patterns to identify vulnerable data flows, resulting in analysing not only the vulnerability of such snippets but also its accessibility.

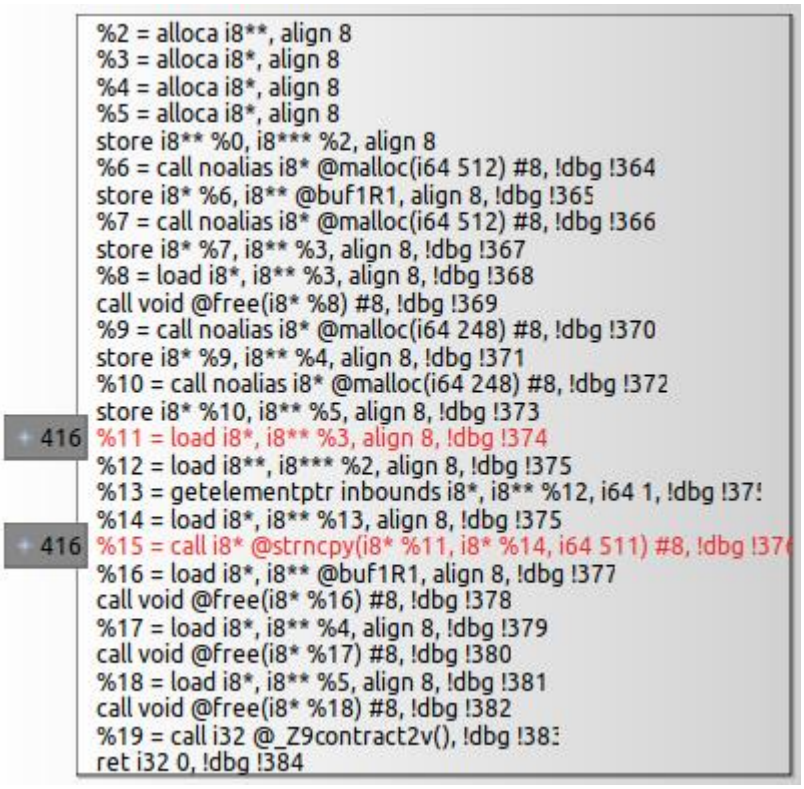


Figure 37: Basic block of test case with vulnerable instructions

Figure 37 shows parts of a test case based on examples included in the Juliet Test Suite for C/C++ 1.3. The test suite contains test cases organized under 118 different CWEs. The test case shown contains the pattern CWE-416. The weakness *CWE-416: Use After Free* describes the use of previously freed memory. This can cause the program to

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

crash, use unexpected values or execute arbitrary code.

The program first reserves memory for a pointer (%3), then loads the pointer into a virtual register (%8) and subsequently releases the memory location (call void @free(i8* %8)). The instructions %11 and %15 then access the freed memory location directly or indirectly, thus satisfying the pattern CWE-416, which results in the analysis marking and classifying the pattern.

2. Score-based Code Weakness Assessment (SCWA)

The Common Vulnerability Scoring System (CVSS) represents a common way of assessing the severity of a vulnerability after a system was exploited. It is a widespread industrial standard. The Score-based Weakness Assessment (SCWA) uses the contextual information obtained from the previous step to simulate the exploitation of the system to derive a CVSS-based score before the system is actually exploited. This is done by mapping architectural features to the CVSS base metrics. The mapping is then used to derive one or more scores based on the mapping quality and reachability of code weaknesses.

This analysis leads to the ranking of code weaknesses in context of a system by the CVSS. As a result, services are tagged with a range of CVSS scores based on the number of contained code weaknesses and their mapping quality and reachability. This helps developers to identify quickly code snippets that are exploitable and subsequently harmful to the system.

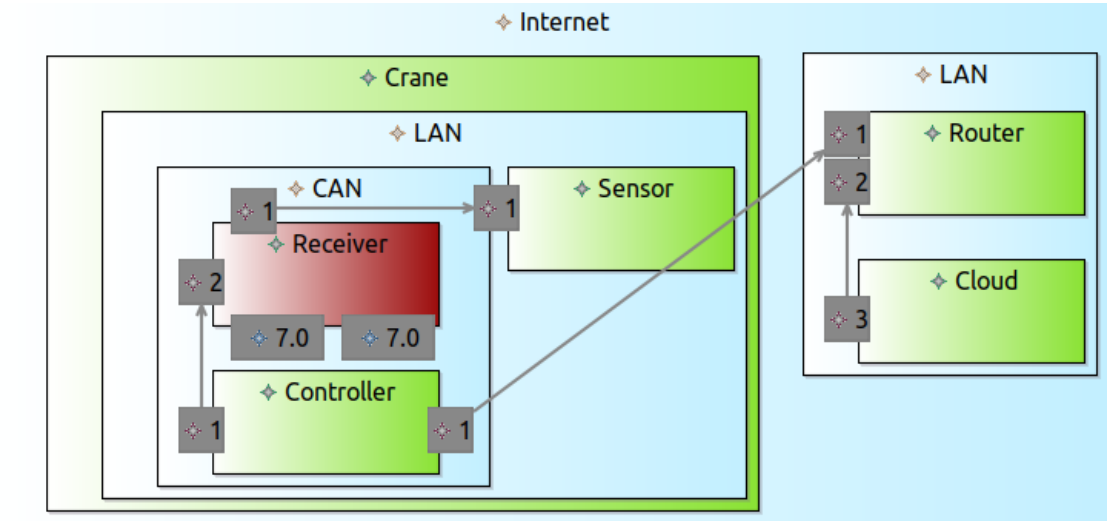


Figure 38: System model of test case with derived CVSS score

Figure 38 shows a simplified version of the system model of the *CPS4EU Use Case 2: Collaborative Lifting*. For evaluation purposes, we defined that the test case shown in Figure 36 represents the functionality of the system component Receiver. In addition, we specified that the function containing the weaknesses is an API function and therefore reachable, since the weakness is always part of the execution when the function is called. The weaknesses found are mapped to a set of CVSS scores along with the input and derived context information from the system model. In this case, the CVSS scores range from 6 to 7.

3. Code Weakness Impact Analysis (CWIA)

The CVSS reflects the severity of a vulnerability contained in a component of the system. The base score of the CVSS is composed of Exploitability Metrics (EM) and Impact Metrics (IM). The EM characterize the properties of the vulnerability that resulted in a successful attack. The IM reflect the consequences that are most directly and predictably associated with the attack. Therefore, the CVSS is restricted to reflect the severity for a specific vulnerability of a specific system component.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

The Common Weakness Impact Analysis (CWIA) assess the effect of vulnerabilities on the system as a whole. A combination of data flow and control flow analysis is used to measure the impact of a code weakness by tracing vulnerable variables and identifying their scope.

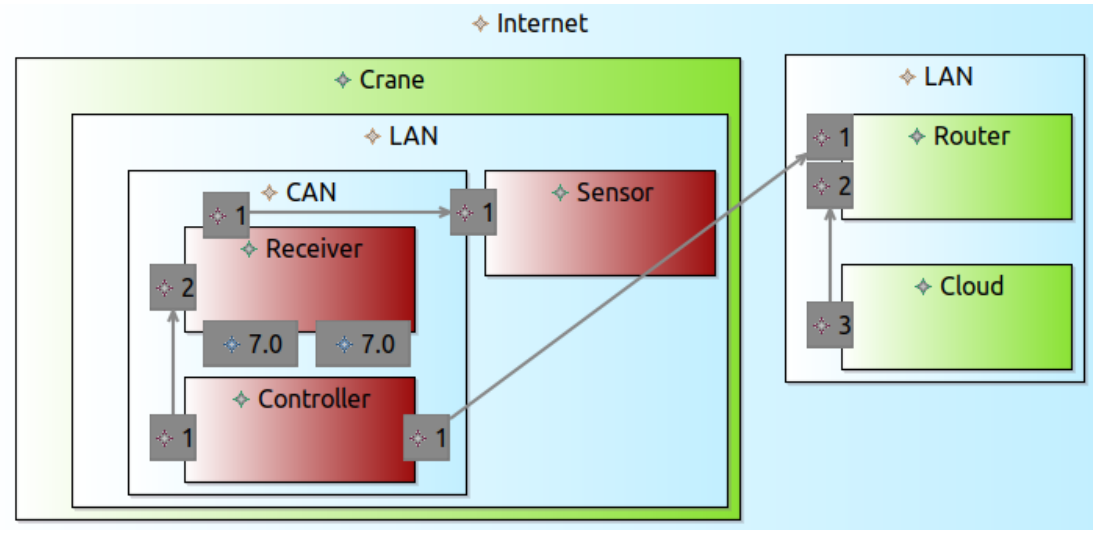


Figure 39: System model of test case with visualized impact of vulnerabilities

Figure 39 shows the expected impact of the previously detected code weaknesses on the system. Since arbitrary code can be executed, any system component that is connected to the vulnerable system component can potentially be manipulated. As a result, the system components Controller and Sensor are marked as flawed.

2.8.1.4 Model-based Analysis

As shown in the overview, beyond the code-based analysis a holistic view on the code and architecture level is necessary and more over on both levels analysis must be done. On the code-based and architectural model first analysis were defined in the last section. In this section, a detailed view on model-based analysis is presented. Let us start with the pattern recognition framework or PRF for short.

PRF – Pattern Recognition Framework:

To detect vulnerabilities in CPS models using patterns in the design phase, a simple definition of these patterns is not sufficient. In addition, grouping, mapping, and final execution must be ensured. These steps are enabled by the UnA's PRF. The main goal is to define and query patterns in a structure. In this phase, UnA focuses on design patterns rather than code patterns to detect code smells. However, design patterns may provide clues to later code problems and UnA's code analyses.

Furthermore, the framework focuses not only on the identification of flaws, but also on the subsequent evaluation and mitigation of them. To this end, categories are incorporated to enable this assessment, but also categorization for pattern selection. CPS-specific or safety- and security-specific categories and elements are integrated. On the one hand, this allows safety and security problems to be specifically addressed, and on the other hand, identification can be seen as a preliminary stage to vulnerability assessment. In this way, for example, it can be quickly decided which design flaws need to be eliminated and with what priority. By focusing on CPS models, specific aspects of the requirements for these particular models can be defined and verified.

The detection of flaws is done by design patterns. They are divided into positive patterns and negative anti-patterns. Patterns describe desirable design decisions that promise an absence of vulnerabilities. In order to detect possible erroneous modelling decisions, model components are searched for that do not correspond to these patterns.

In this search, the model components are examined for conditions of the unique pattern definitions. Only if all pattern conditions are violated, a match is indicated. Anti-patterns define dangerous design decisions that should

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

not exist or have caused problems in the past. For anti-patterns, automatic flaw detection looks for model components that match them. The framework makes it possible to define generally valid patterns that are suitable for all CPS models. However, specifically designed patterns for individual CPS models are also possible and necessary. To cover different levels of abstraction, patterns can be defined at the metamodel level or at the model level. Depending on whether generic patterns or instance-specific patterns are to be defined.

Listed below are examples of pattern or anti-pattern to detect safety flaws created by UnA to analyse CPS4EU use cases.

Hazard	Devices with a direct influence have a zero tolerance to failures.
Pattern / Anti-Pattern	There must be a design diversity in the system, which is guaranteed by a different backup device. In addition, zero tolerance devices can only be connected to the Internet via gateways to reduce the amount of possible dangerous impacts.
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*PhysicalEntity, HumanEntity, PhysicalConnection*
Node2Relation	*connections*
NodeAttribute	*direction: SourceToTarget, trustLevel: low, type: machine*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*Network, VirtualEntity, PhysicalEntity*
Node2Relation	*Entities, connection*
NodeAttribute	*networkType: internet; type: gateway; type: machine, trustLevel: high*

Hazard	Measured values are subject to fluctuations caused by different environmental conditions.
Pattern / Anti-Pattern	The system must not perform measurements without device location identification.
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*System, Network, PhysicalEntity, Services*
Node2Relation	*network, entities, services*
NodeAttribute	*type: sensing*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*GlobalCoordinate, LocalCoordinate*
Node2Relation	*entities, referencepoint*
NodeAttribute	*GlobalPositionProtocoll: GPS, coordinate != null*

Hazard	Improper handling of data units and metrics.
Pattern / Anti-Pattern	*For each data, record a compatible data unit and metric must be modeled.*
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*Service*
Node2Relation	*/*
NodeAttribute	*serviceType: sensing*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*Service, Metric, Constraint*
Node2Relation	*metrics, constraints*
NodeAttribute	*metric == action, constraint == servicetype*

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Hazard	*Machine is active although worker is to close*
Pattern / Anti-Pattern	*Location of Human and Machine must be distant by each other*
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*HumanEntity, PhysicalEntity*
Node2Relation	*network, entities*
NodeAttribute	*/*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*HumanEntity, PhysicalEntity, GlobalCoordinate, LocalCoordinate*
Node2Relation	*network, entities, location, referencepoint*
NodeAttribute	*humanEntity.coordinate != PhysicalEntity.coordinate*

Hazard	*Complicated Authentication in urgent situations*
Pattern / Anti-Pattern	*A complicated 2-way authentication is implemented in a service allow a emergeny stop*
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*Service, MachineEntity*
Node2Relation	*entities*
NodeAttribute	*serviceType: acting; type: mobilePhone*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*HumanEntity, Service*
Node2Relation	*entities*
NodeAttribute	*stakeholder: ambulance; authenticationType: 2way*

Listed below are examples of pattern or anti-pattern to detect security flaws created by UnA to analyse CPS4EU use cases.

Threat	*Bluetooth-enabled device is manipulated and prevents the correct execution of services.*
Pattern / Anti-Pattern	*Devices with a high SIL must only allow Bluetooth 4.0 or newer, as this does not allow an unlimited number of authentication challenge requests or no encryption*
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*MachineEntity, Network, Connection*
Node2Relation	*network, connections*
NodeAttribute	*SIL: high*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*PhysicalConnection, Encryption*
Node2Relation	*encryption*
NodeAttribute	*protocol: bluetooth4.0; encryptiontype= SAFER* + block cipher

Threat	*Eavesdropping by communication from external to internal devices.*
Pattern / Anti-Pattern	*High distance communication must be implemented with LowRaWan as communication protocol to ensure secure coupling.*
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*MachineEntity, GlobalCoordinate, Network*
Node2Relation	*entities, location, network*

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

NodeAttribute	*MachineEntity1.coordinate != MachineEntity2.coordinate, network1 != network2*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*PhysicalConnection*
Node2Relation	*connection*
NodeAttribute	*type: LowRaWan*

Threat	*Physical attacks conducted through malicious node injection.*
Pattern / Anti-Pattern	*Each critical node must be registered in the identity store and connected as a peered device to its respective gateway.*
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*MachineEntity*
Node2Relation	*/*
NodeAttribute	*state:critical*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*Workflow, Step, Service, VirtualEntity*
Node2Relation	*workflows, next, services, entities*
NodeAttribute	*message: name; next: registration; serviceType: identifying; virtualEntity: cloudGateway*

Threat	Unauthorized access to a resource.
Pattern / Anti-Pattern	Access services must have a connection assigned to a user with appropriate role.
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*Service*
Node2Relation	*/*
NodeAttribute	*type: access*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*HumanEntity*
Node2Relation	*entities*
NodeAttribute	*Stakeholder: SecurityTeam, role: operator*

Threat	Critical services can be triggered by remote control.
Pattern / Anti-Pattern	Services on resources with high risk and low trust level must not be remotely controllable with another device.
Implementation	*Supercategory: Required "Node X Attribute X Relation" Combination*
Element Filter Condition	*Defines condition(s) to select the specific concerned elements*
Node2Node	*MachineEntity, Service*
Node2Relation	*entities, subentities*
NodeAttribute	*entity1!=entity1; entity1.trustLevel: low; servicetype: acting*
Pattern Requirement	*Defines requirement(s) for pattern recognition process*
Node2Node	*Network*
Node2Relation	*network*
NodeAttribute	*entity1.network == entity2.network*

To assess the identified flaws UnA has developed four model-based analyses, which are conducted before UnA's code-based analyses. They are working in a continuous workflow, as described in earlier deliverables.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Failure Impact Analysis (FIA)

After identifying a vulnerability, the next step is to analyse the impact on other elements in the event of an accident or attack due to the vulnerability. This is what an FIA aims to do by identifying the probability distribution of the effects. By recognizing patterns, the causal relationships of the flaw causes are already known. Accordingly, a top-down approach to the system is not required. A bottom-up approach to monitoring the impact on elements that are logically dependent on the faulty element must be used. This, in turn, provides information on which components could be negatively affected by a flaw to a high degree. The FIA uses concepts of Bayesian Belief Nets and provides an assessment of the conditional probabilities of the faulty elements and their associated nodes and relationships. Therefore, a final assessment of weakly or severely impacted connected elements can be made. Severely impaired elements require further assessment or mitigation.

This analysis consists of FIA elements (see Figure 40), each corresponding to an element from the CPS model. They are assigned to layers. Each element has, besides its independent probability value, joint probability tables and condition probability tables for the BBN calculations. In addition, they have matrices to display the results of the flaw effects.

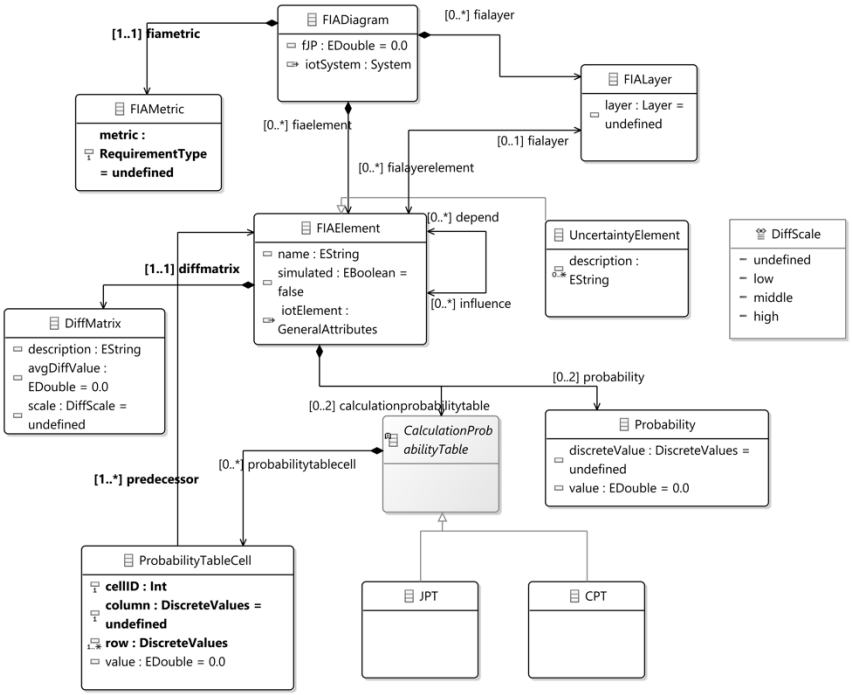


Figure 40: FIA meta-model

Thus, it is possible to perform a failure impact analysis on all levels of abstraction and obtain traceability of the failure impact. Details can be found in the PhD. thesis of Julia Rauscher.

Quantitative Impact Analysis (QIA)

QIA aims to estimate the cost of impact and therefore does not focus on the technical aspects of a flaw. Rather, it focuses on the upper layers and the financial burden to assess the urgency and usefulness of countermeasures to correct flaws based on the number of flaws and the likelihood of their occurrence. Elements identified as highly impacted based on FIA performance can be analysed for quantitative impact, as the current architecture situation must account for potential costs in the event of an accident or attack occurring. To do this, the costs are quantitatively calculated in an impact diagram. QIA is performed using the calculated conditional probabilities of the affected elements and their connections to estimate the frequency of occurrence and possible cumulative costs.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

QIA performs multiple estimates for individual elements, entire layers or systems, or potential severity.

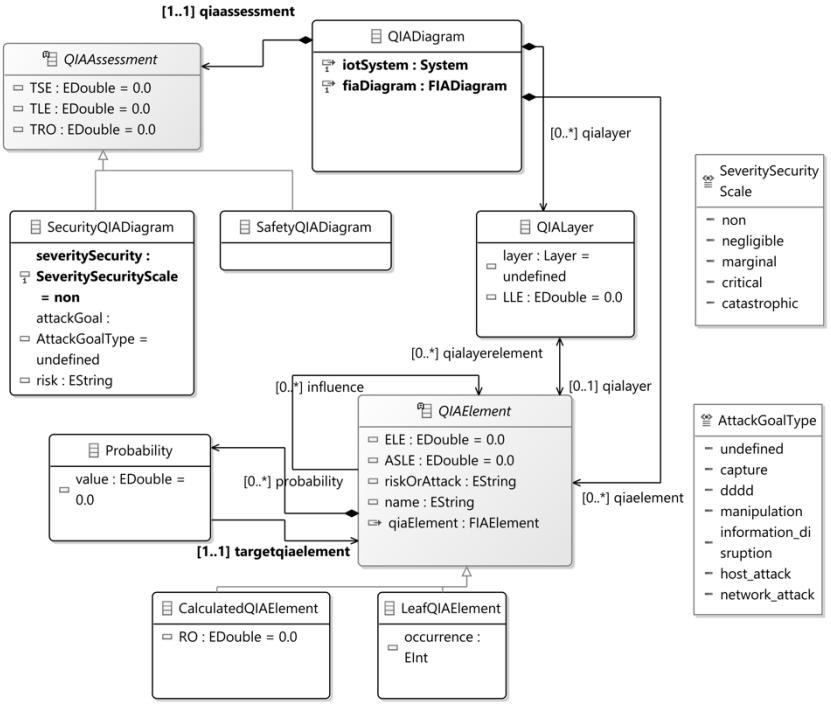


Figure 41: QIA meta-model

An example for such a QIA analysis is shown in Figure 41.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

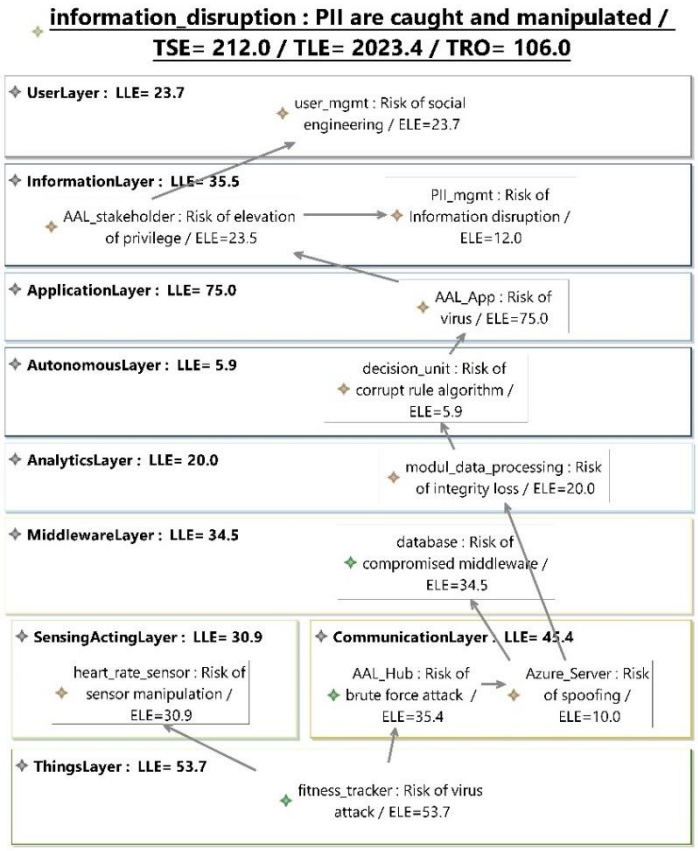


Figure 42: QAI example

Countermeasure Decision Support Analysis (CDSA)

Once vulnerabilities have been identified and assessed, the current as-is situation must be modified in a future to-be architecture to mitigate or address these vulnerabilities. Since there are several possible target architectures, the CSDA can be performed to compare different scenarios and find the most suitable architecture for the use case. The goal of a CDSA is to review the necessary countermeasures to mitigate or prevent the identified flaws and potential impact. Since various new design scenarios are possible, they must be weighed against each other and against other countermeasure options. Therefore, different countermeasures spanning multiple levels can be defined and assigned target requirements for countermeasures. These various requirements are balanced against the vulnerabilities and countermeasures. The weighted countermeasures can be analysed and compared to provide the evaluated possible To-Be scenarios.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

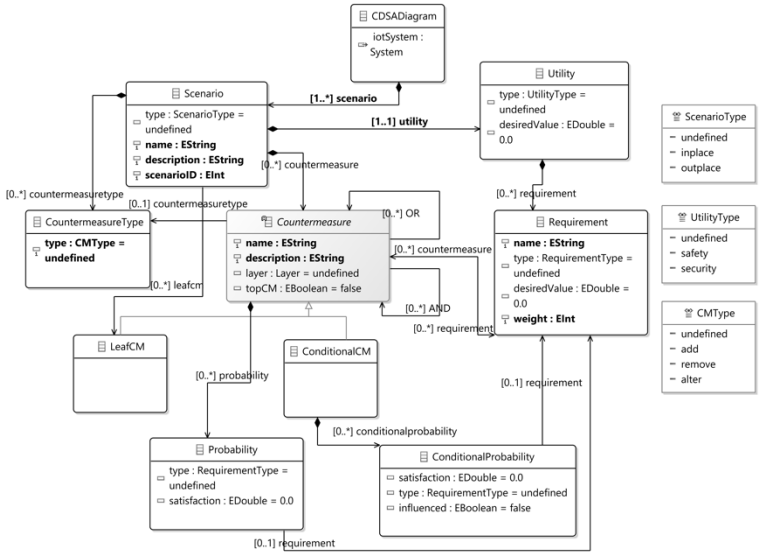


Figure 43: CDSA meta-mode

An example for CDSA is shown in the following figure:

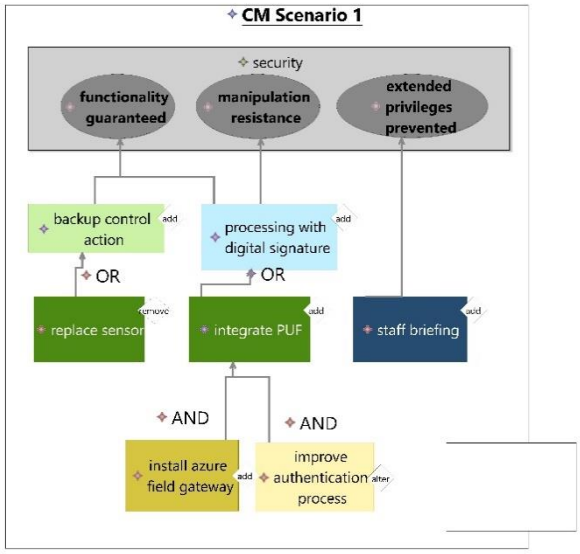


Figure 44: CDSA example

Service Interoperability Analysis (SIA)

A new target architecture also includes elements of the previous architecture. Therefore, the previous services and the services of the new architecture must be examined for the interoperability of these two groups of services. SIA aims to compare an old, faulty, as-is scenario with previous services with a new to-be scenario with new services. This allows an assessment of whether the newly planned countermeasures would introduce new critical services and thus create new problems or prevent proper interaction of current services. Services can be assessed for individual services in the overall system, for the matching of a pair of services, and for the so-called power set of service, which analyses the setup of all connected services. The evaluation approach is based on the predefined requirements for the quality of services with probabilities and flexible comparison values that fit the pairs of services

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

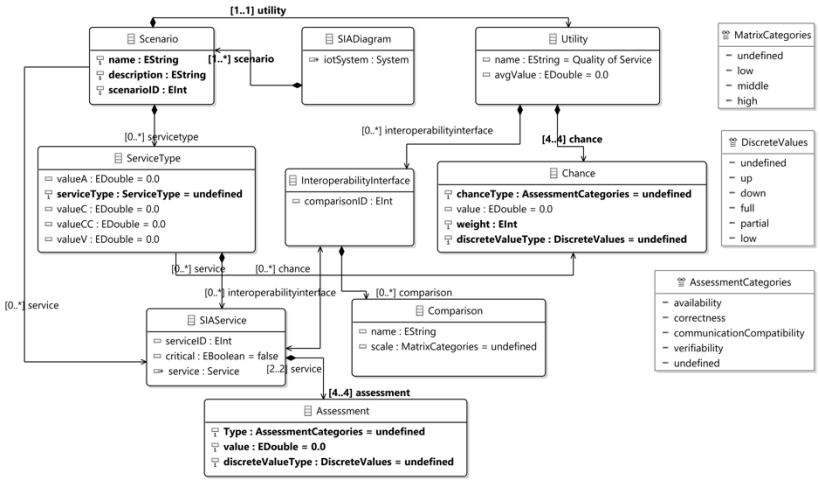


Figure 45: SIA meta-model

An example for SIA is shown in the following figure:

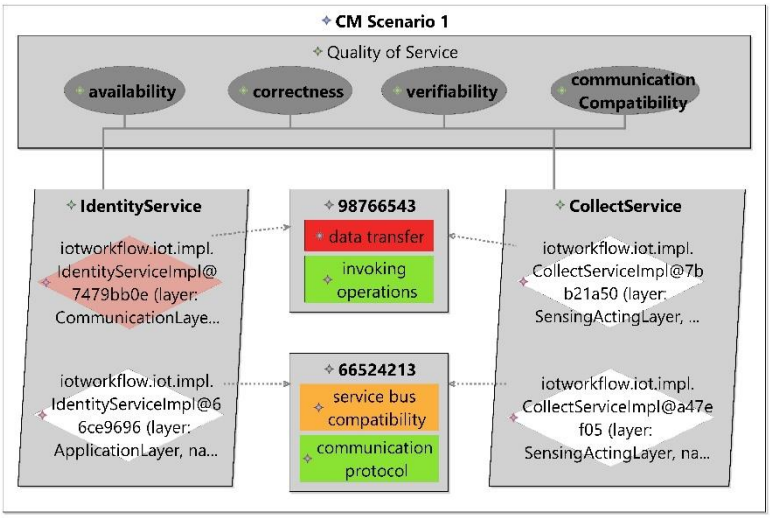


Figure 46: SIA example

2.8.2 Cluster

Initially, UnA participated in the iterative code optimization cluster but due to EMX reducing its effort the cluster was cancelled and UnA joined the heterogeneous co-simulation cluster. This cluster focuses on integrating simulation components generated from different tools and carefully studying the interactions between these tools. However, the simulation components are written in C++ and are therefore suitable for security analysis.

The focus is on identifying weaknesses in the simulation component code and assessing their impact on the system as CPS suffer from large attack surfaces.

UnA carried out the following steps:

Modelling of the system model for the hybrid vehicle use case (*Figure 47*).

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

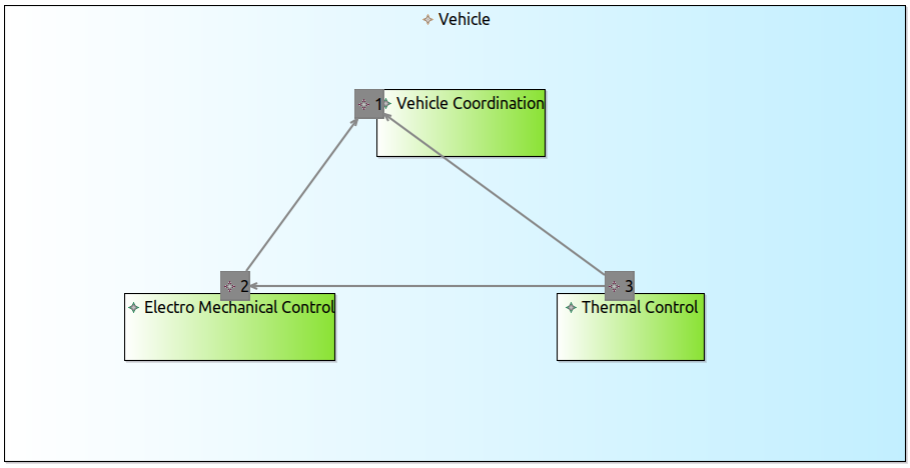


Figure 47: System model of the hybrid vehicle use case

Analysing the source code of the control system (Figure 48). No weaknesses were discovered.

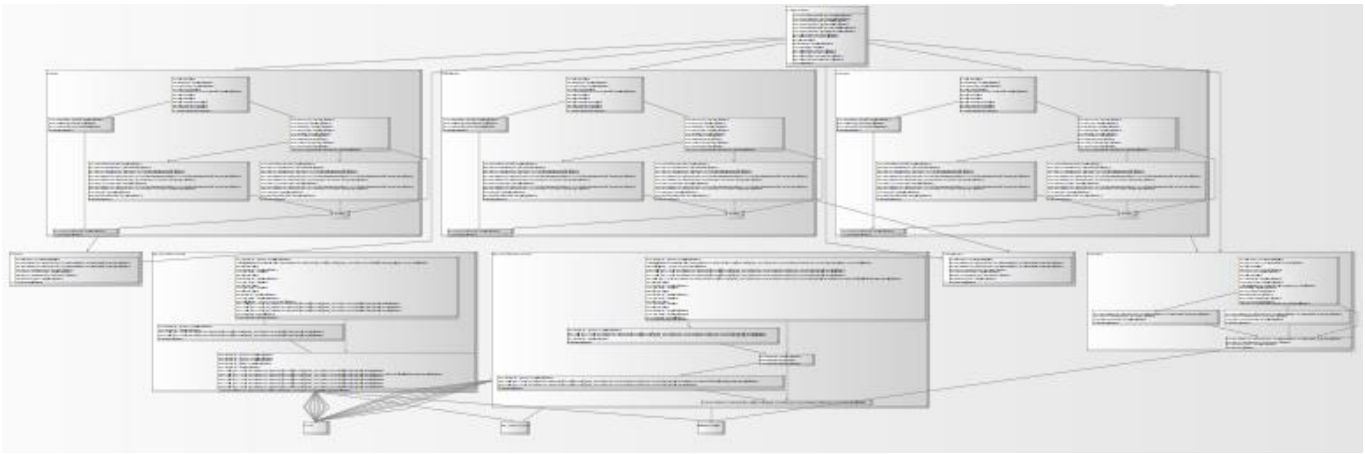


Figure 48: Part of code model of control system software

Moreover, the above-defined pattern and anti-pattern can also be applied to the cluster use cases as well as the described architecture analysis.

2.8.3 KPI

Question	Proposed success criterion/criteria	Result
Concept for analysis of safety concerns (safeconcs)	Success = {Successful application of 3 pattern and anti-pattern for PIARCHs} AND {4 applicable architecture analyses to identify errors and shortcomings of the architecture}	5 pattern and anti-pattern for PIARCHS were defined and 4 applicable architecture analyses to identify errors and shortcomings of the architecture were developed (PRF, FIA, QIA, CDSA). **Success**
Concept for analysis of security concerns (secconcs)sensors measurements	Success = {Successful application of 3 pattern and anti-pattern for PIARCHs} AND {4 applicable architecture analyses to identify errors and shortcomings of the architecture} AND {1 applicable code analysis to identify errors and shortcomings of the code artefacts}	5 pattern and anti-pattern for PIARCHS were defined, 4 applicable architecture analyses to identify errors and shortcomings of the architecture (PRF, FIA, QIA, CDSA) and 3 applicable code analysis to identify errors and shortcomings of the code artefacts were developed (SCA on IR, SCWA, CWIA). **Success**

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

Table 7: UNA KPIs decomposition

2.8.4 Lessons Learned

In the CPS4EU project, we were given the opportunity to evolve our safety and security analyses by applying our ideas on use cases provided by industrial partners. We developed different tooling and were able to work with industry partners to improve and evaluate it benefitting from their experience. In particular, the different views on the safety and security issues from different partners allowed us to identify real-world problems and had therefor the possibility to increase our knowledge to support the development process with applicable tools and methods, which are not based on theoretical considerations but by problems of the industrial partners.

2.9 ITI

ITI develops R&D applied to the needs and problems of companies, looking for technological solutions that respond to social and economic challenges, improve industrial competitiveness, promoting a more intelligent and sustainable society. The Cyber-Physical System team is developing art2kitekt, a web-based tool that can be used for modelling Cyber-Physical Systems (CPS). It includes requirements management and provides several analysis techniques, simulation, and monitoring services to ensure the system's feasibility in the early phases of the product life cycle. This application allows defining the hardware platform and the software application to characterise the system. The thread structure of the implementation can be automatically generated based on different operating systems.

2.9.1 Objectives

The ITI objective is to reduce the integration effort in CPS systems, improving both interoperability and reusability of the simulation components. To achieve this, ITI studies the possible interactions between the tools and methodology to develop, adapt, and integrate the simulation components.

2.9.2 Innovations

ITI has developed new features to art2kitekt (a2k) to model and simulate systems using a standardised format, allowing us to offer data exchange and synchronisation services with external tools.

During the project, the tool has been extended to offer the following features:

- Dispatch messages between different simulation units. ITI has integrated an RTI (Runtime Infrastructure) module to route the information between nodes.
- Access to the functions provided by the RTI. ITI has created an adapter to facilitate the use of the HLA services through the RTI.
- Control the execution of the simulation. ITI has developed a master to manage the simulation execution.
- Viewers and analysers that show the evolution of the simulation and results.

On the other hand, ITI has also developed a code generator that creates the source code that facilitates the adaptation between standards (FMU and HLA).

2.9.3 Experimental evaluation

This section will demonstrate the innovations described in the last section. To do that, we tested all the components developed through HLA simulations to demonstrate our innovations. As we described in the last deliverable, in the HLA standard, a distributed simulation is called a federation, which comprises several HLA simulation entities, called federates. These units are physical, mathematical, or logical representations of processes and systems. These federates are connected to the Run-Time Infrastructure (RTI) to interact.

A classical HLA federate consists of a simulation model and a local RTI component (RTI ambassador). The RTI ambassador eases the integration of the simulation modules allowing the federate to exchange information into the federation and request the services that the RTI provides.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

As explained in previous deliverables, the modelling tools can export co-simulation FMU/FMI models. These models comply with the Functional Mock-Up Interface (FMI), a standard that includes an interface for the dynamic exchange of FMU models. It is a compressed file containing the Model Description file in XML format (modelDescription.xml), the Compiled library with the model, and in some cases, the source code.

Adapting the FMU to the HLA federate is a generic process that can be automated. Therefore, we developed an automatic HLA Code Generator tool that allows source code creation, which is the glue between the FMU and the HLA standards.

The group has run an inverted pendulum to test all the components developed in a2k. Some simulation units are FMU, which we created with Simulink and Open Modelica. Therefore, we use the Code Generator tool to create the HLA units from the FMU units.

Inverted pendulum:

The Inverted pendulum simulator aims to test the HLA integration with a2k, and the federate generation using the FMU units as an input. The inverted pendulum consists of four simulation units: a control system, a pulse generator, the inertial measurement unit (IMU) and the pendulum dynamics. The pulse generator produces signals that destabilize the pendulum. The control system tries to keep it balanced, and the IMU simulates the sensors. The pendulum dynamics unit has differential equations representing the pendulum's time evolution. We can manage the simulation with a2k using the simulation and interface with start and stop options and connect external viewers and checkers to the RTI. The checker's purpose is to analyse the correct operation of the simulation elements. Figure 49 shows the connections between the units and the simulation master with the RTI.

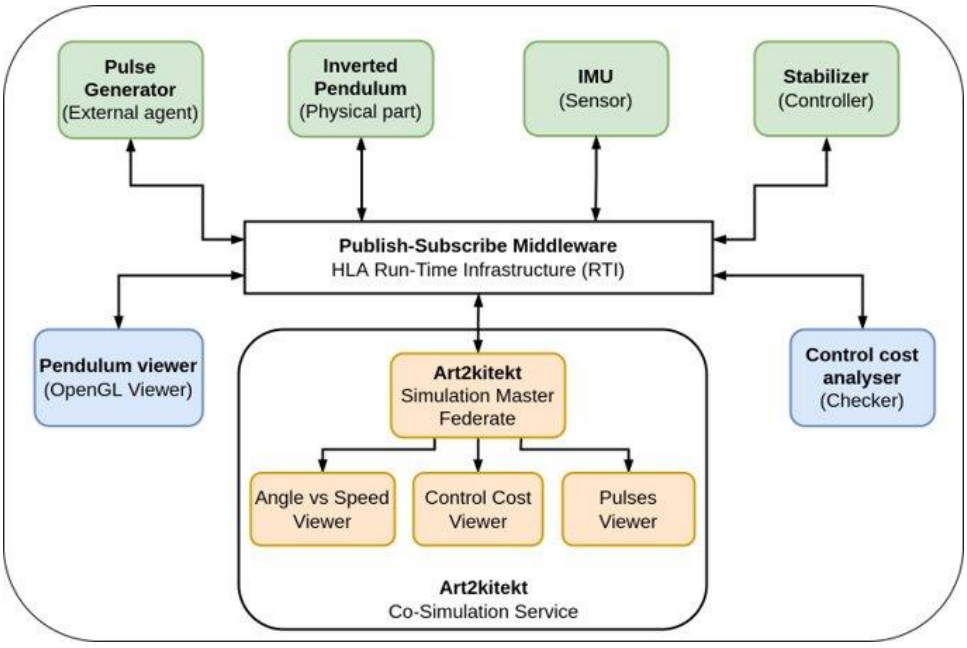


Figure 49. Inverted pendulum simulation components diagram

With a2k, we can manage the simulation. A component diagram (Figure 50) shows the federates joined to the simulation and the data relationship. Additionally, this monitor checks the status of each federate during the whole execution and ensures the connection remains established.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

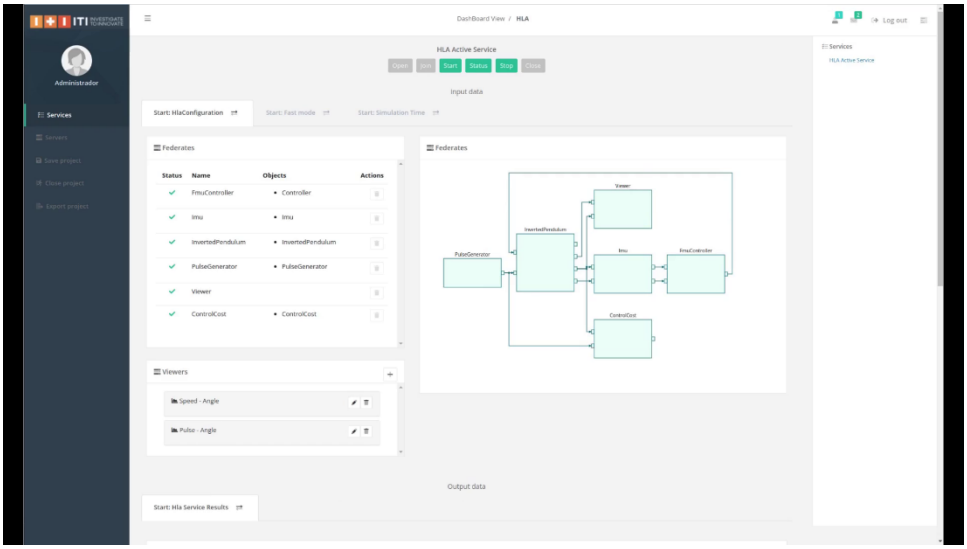


Figure 50. a2k web interface for simulation service

The simulation starts once all federates have joined the simulation. This action (Play) is carried out from the commands available at the top of the service interface (Figure 51). The Stop action forces the simulation to stop early during the execution, before the defined simulation time is up.

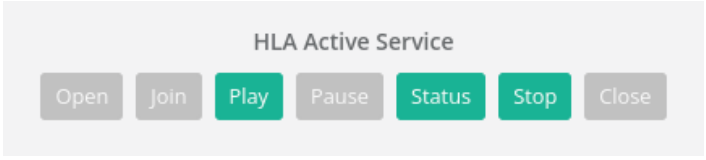


Figure 51. HLA federation actions components in a2k

The data exchanged by the different federates, can be displayed in the a2k simulation service during the execution. To do this, a2k provides several chart components that can be used to observe the data. (Figure 52).

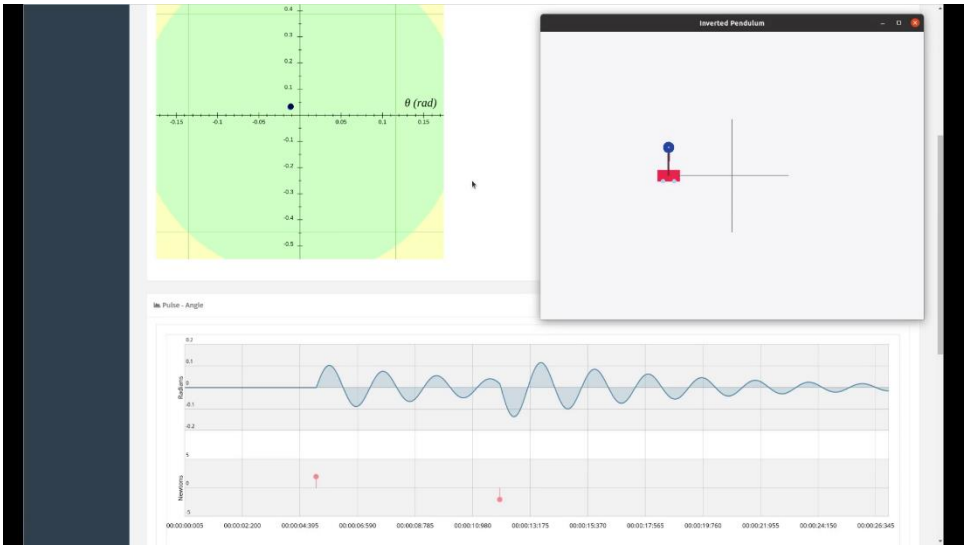


Figure 52. a2k chart components

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

2.9.4 KPIs

Question	Proposed success criterion/criteria	Result
How can the time effort in the design stage be reduced by using different modelling tools in the same simulation environment?	Reduction of CPS design time by 30% thanks to the use of HLA as a standard run-time interface solution.	As it is measured during the experimental validation, the average time needed to design a simulation component again with another modelling tool is 4 hours. Conversely, the average time needed to adapt the same simulation component to be executed in an HLA simulation using the adapter is 1 hour. The average time reduction achieved is 1– 1/4 = 0.75 à 75% > 30%
How many models, generated by different tools, have been successfully integrated to be executed into an HLA simulation using the HLA adapter?	At least three pre-existing modelling tools should be integrated and executed into an HLA simulation.	Thanks to the HLA adapter provided by ITI, the following models have been integrated and executed in an HLA simulation: - Standalone FMU/FMI models generated by Simulink. - Tool-Coupling FMU model using Simulink. - FMU/FMI models generated by Modelica, - FMU/FMI models generated by Twin Builder from Ansys. Therefore, four modelling tools have been integrated.

Table 8: ITI KPIs decomposition

2.9.5 Lessons Learned

In the CPS4EU project, our group has had the chance to learn and integrate distributed simulation technologies into our tools. The group gets the knowledge to provide services and tools needed to deliver the most innovative interoperability solutions and offer the management, guidance, and technical support that the industry needs to integrate distributed simulation systems into interoperable solutions successfully.

We had the opportunity to work with key companies in this sector, such as Sherpa and Ansys, through stimulating use cases, where we learned methodologies to simulate complex scenarios.

The group had essential benefits using two popular standards: FMI and HLA. We could create a new application that allows generating HLA simulation units from FMU units created by external tools. Therefore, we reduce the time needed to create distributed simulations using external tools, such as Simulink or Open Modelica.

The Federation Object Model (FOM) specifies the Object Classes and Interaction Classes used to exchange data, making it an up-and-coming technology to integrate our real-time services in distributed simulations.

2.10 UGA

2.10.1 UGA Technology

2.10.1.1 BIP Framework

VERIMAG has developed the BIP (Behaviour, Interaction, Priority) component framework for about 12 years. BIP has a formally defined operational semantics, which underlies all the analysis, transformations, and implementation techniques and, therefore, supports the development of rigorous model-based design flows. Component-based systems in BIP are modelled by composing atomic components (the behaviour as extended automata with code) with multiparty interactions and restrictions through dynamic priorities. Recent research expands the original

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

framework to increase its adoption for additional categories of systems and/or application domains featuring real-time and stochastic systems. The characteristics of the BIP framework are as follows:

1. Component-based, providing a family of operators for building composite components from simpler components,
2. Model-based, describing all software and systems according to a single semantic model, explicitly modelling architecture and interactions between the various components,
3. Tractable, guaranteeing correctness by construction and thereby avoiding monolithic a posteriori verification as much as possible.
4. BIP models are amenable to verification through statistical model checking tool called: SMC-BIP that accepts properties expressed in PBLTL temporal logic as portrayed in Figure 53.

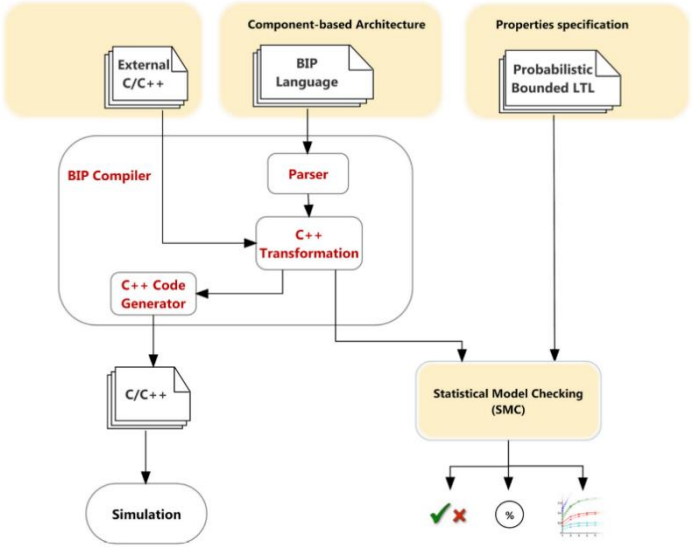


Figure 53: SMC- BIP approach

2.10.1.2 Model-To-Model transformation

Integrating UML and BIP is an appropriate way for the rigorous development of complex and critical systems. On the one hand, UML is a standard graphical notation with a visual and structural aspect through its diagrams. On the other hand, BIP is a textual representation that allows for building formal models with the support of external code for specifying component behaviours.

Meanwhile, designing using UML requires first modelling the system using composite diagrams that mainly handles the elements of constructs handled by BIP such as ports. The transformation approach targets the UML State Machine Diagrams in the BIP automata's semantic. The generated model is then checked using SMC-BIP.

Table 9 defines the main mapping rules to translate the basic structures needed for our case study from UML to BIP. Work in progress considers other structures such as the priorities of interactions between components. A prototype is developed to automate the translation using the Eclipse Acceleo Tool.

UML	BIP
Composite component	Compound
Atomic component	Atom
Connection	Connector
State	Place
Transition Event	External Port/Internal Port
Transition action	BIP expression
Transition guard	BIP guard expression (provided)
Variable	BIP Data

Table 9: Mapping rules from UML to BIP.

The UML composite components become BIP compounds, and the UML atomic components become BIP atoms. BIP connectors represent connections between UML components. State machine diagrams specifying the

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

behaviour of UML atomic components are also translated into BIP specification attached to the corresponding BIP atoms. States are specified by BIP places and transition events by BIP ports. Then, we associate the transition actions and transition guards to the BIP ports. The published paper in [1] gives more details and descriptions.

2.10.2 Experiments: Modelling, simulating, and monitoring WIKA protocol for collaborative lifting

2.10.2.1 Modelling

We present an approach that combines the SEStools from TUC, the BIP Framework from UGA and the THEMIS tool from INRIA for modelling, simulating, and monitoring Cyber-physical systems. We apply the proposed approach for collaborative lifting use case from WIKA.

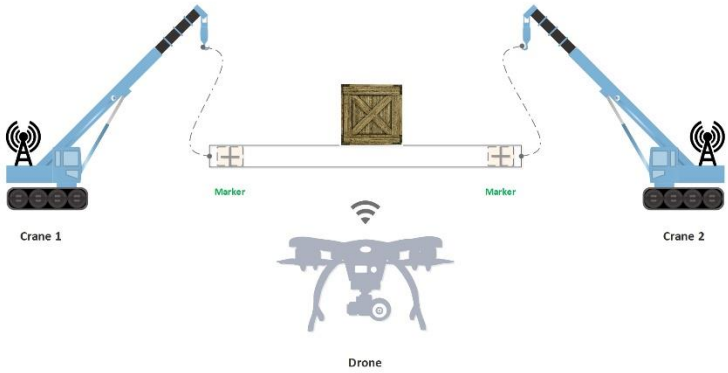


Figure 54: Drone-Crane Orchestration

As shown in Figure 54, BIP can be used to specify high-level model scenarios for a given protocol and output useful traces that can be monitored and verified by THEMIS (INRIA). Furthermore, since we are given multiple sensor data from WIKA and TUC Drone (s), combined with generated trajectories from WIKA, there needs to be an agreement between sensors which is not trivial in addition to computation to match trajectories; the protocol for agreement along with the threshold for errors can be modelled in BIP after which it generates a trace that THEMIS can monitor. Optionally, partners can provide the traces immediately to THEMIS for monitoring in case the traces are simple. THEMIS utilizes state-based decentralized information with discrete-time; for each component, a file must be provided where each line represents a timestamp, and the line contains atomic propositions and their Boolean values at that time (e.g., crane1_moving:t, crane2_moving:f, sensors_correlation_safe:t). Additional input to THEMIS can be provided in terms of sensors and thresholds when needed, as THEMIS has already been used to monitor smart apartments with various sensors. We note that possibly, input can be directly fed to THEMIS monitors by writing a custom "THEMIS Bootstrap" component and "Peripheries" which are input streams for monitors.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

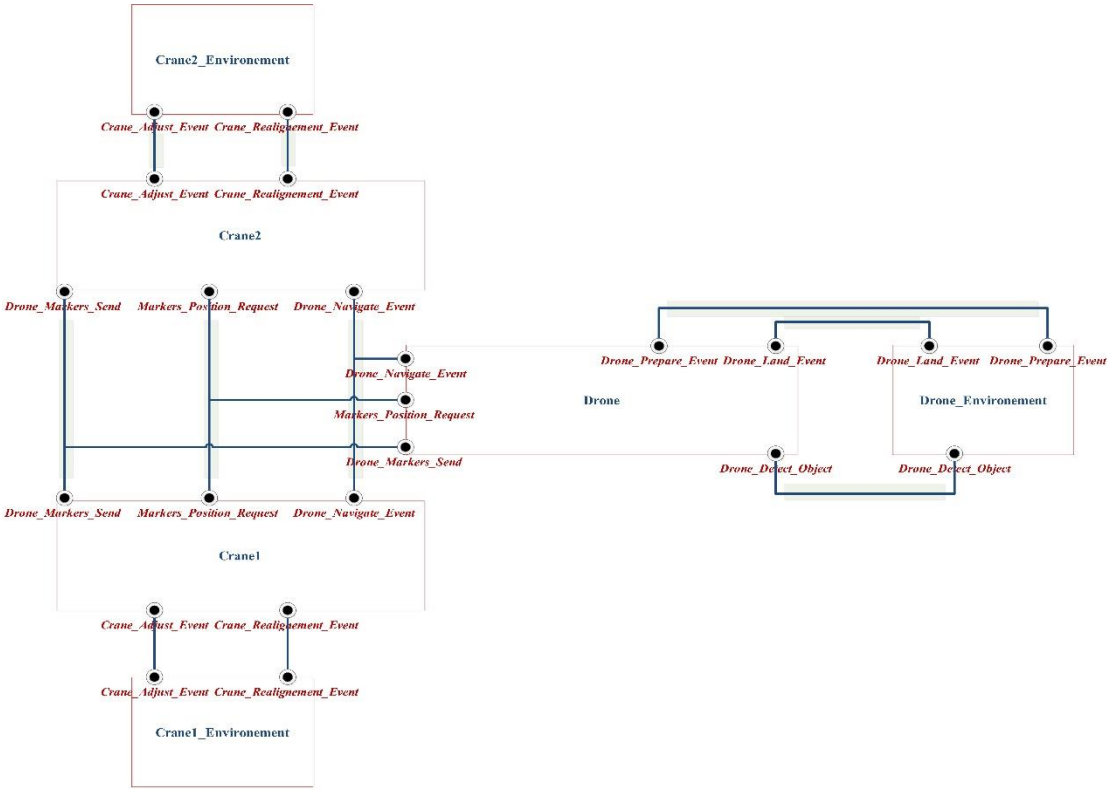


Figure 55: Drone-Crane Architecture

The BIP model that reflects the Drone-Crane orchestration in UML is portrayed in Figure 55, where six components are modelled Drone, Drone_Environement, Crane1, Crane1_Environement, Crane2, and Crane2_Environement. The Drone component is endowed with three export ports "Drone_Prepare_Event", "Drone_Land_Event", and "Drone_Detect_Object". "Drone_Prepare_Event" triggers the execution of the Drone, "Drone_Land_Event" performs the landing, and "Drone_Detect_Object" performs object detection while it synchronizes with the Drone environment that simulates the surrounding area of the drones. The drone component also receives requests from Crane through ports: "Drone_Navigate_Event" and "Markers_Position_Request" to navigate to a specific position and perform markers positions collection, respectively. The Marker's position is transmitted to the Crane through the port "Drone_Markers_Send". The Behaviour of the Drone and its environment is modelled in Figure 56. The Crane is endowed with five ports that mainly send orders to drones through the ports "Drone_Navigate_Event" and "Markers_Position_Request" while it receives markers position through the port "Drone_Markers_Send". Simulating the environment requires to interact through the ports "Crane_Adjust_Event" and "Crane_Realignement_Event". The Crane behaviour and its environment are modelled in Figure 57

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

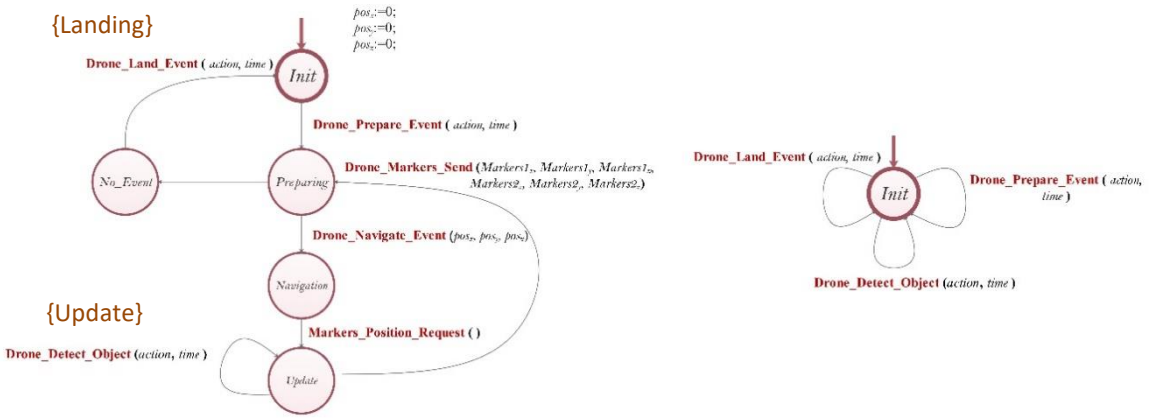


Figure 56: Drone and Drones environment models in BIP

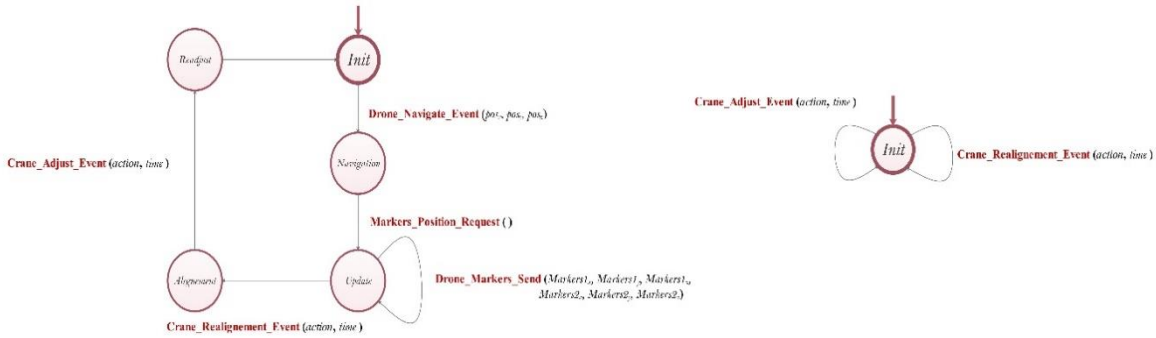


Figure 57: Crane and Crane environment models in BIP

2.10.3 Evaluation

To evaluate our BIP model, we perform statistical model checking using SMC-BIP regarding properties expressed in PBLTL as portrayed on the flow of Figure 53.
First, we want to check the model for the case where two types of communication styles: synchronized and broadcast communication:

RQ 1: What is the probability that both cranes receive the object markers globally during their execution? c1 refers to the first Crane, and c2 refers to the second Crane. “m” refers to the marker value 0 or 1.
Pr=?[G{100} (c1.m==c2.m)]

The results related to both connector's structures are portrayed below and assert that synchronization using synchronized connectors is required for reliable communication and service delivery. The broadcast connections output multiple paths necessary for the modelled systems: Cranes will broadcast requests to drones unavailable at the required time. Therefore, synchronized communication will allow communicating entities to exchange data based on their availabilities. This interaction will reduce the unnecessary paths that are not relevant for the analysis and point out the flaws that may happen at the cranes or drones. Indeed, the functionality of the Cranes is not subject to flaws but deserves interest to locate the malfunctioning.

1. Broadcast connectors	verdict = 36%
2. Synchronized connectors	verdict = 100%

RQ 2: What is the probability that the Drone receives the Marker positions and finally performs a landing? “d” refers to the Drone, and t refers to the time elapsed to complete the landing
Pr=?[(d.Update ==True) U{100} (d.Landing ==true & t<=k)]

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

The results related to the checking of the property RQ 2 are portrayed in Fig 58, where the probability increases as the time increases since the state reaching the landing are more reachable after five times units than in a one-time unit. Noting that the communication style is based on synchronized communication that reduces the irrelevant states and transition inspected by the engine in models in Figure 55, Figure 56, and, Figure 57. The elapsed time "t" is added at the model level by integrating an integer value that is incremented as the transition is performed. Indeed, there will be a gap between the modelled system at BIP and the deployed one. The communication will reuse the communication styles supported by drones and cranes like TCP sockets for synchronized communication (i.e., UDP sockets for broadcast communication style) and more flaws stemming from drones due to the wireless communication. The communication support will hinder the data flow, and the drone may enter sleeping mode. This behaviour shall be handled by retriggering the request from the cranes.

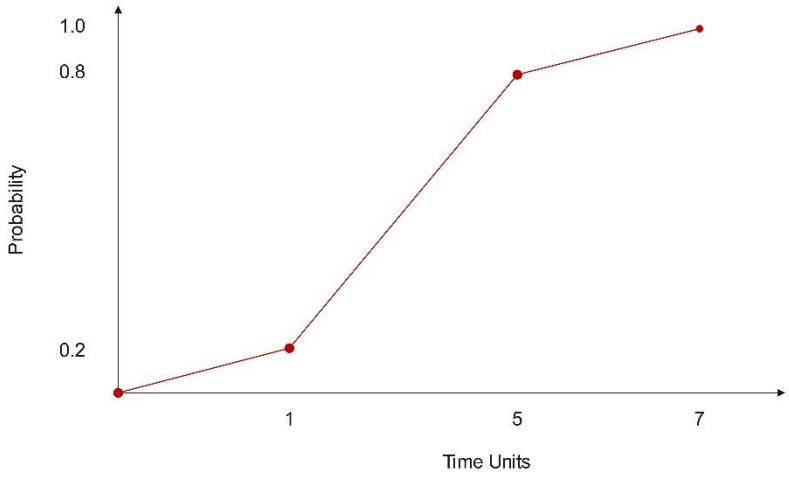


Figure 58: Graphical representation of the results checking RQ2

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

3 EVALUATION OF THE WORK DONE PER CLUSTER

After reviewing the different partners' contributions and describing for each of them through KPIs what was gain from that project, we propose to setup the same kind of view at the level of each cluster.

3.1 HETEROGENEOUS CO-SIMULATION

3.1.1 Introduction and purpose

As described in deliverables 5.4 and 5.5, the Heterogeneous Co-Simulation cluster focuses on integrating simulation components, generated from different tool eco-systems, in distributed simulations.

This cluster aims to reduce the integration effort in a heterogeneous co-simulation, improving both interoperability and reusability of the simulation components. To achieve this, we study the possible interactions between the tools involved and the methodology to develop, adapt, and integrate the simulation components.

To demonstrate these properties, we simulate the hybrid vehicles (PHEVs) described in deliverable 5.5, section 2.1. Figure 9 shows a cluster adaptation for the hybrid vehicle use case. Each of these components is responsible for simulating a specific characteristic, and all together, they form a complex simulation (interoperability). Physical parts include the vehicle model and thermal and electro-mechanical systems and are modelled by the Operating System federate, the Lead Temperature federate, and the Leaf Electromagnetic federate. The vehicle coordination and the electro-mechanical and thermal control are defined by the Control System federate. Finally, the Scenario federate represents the simulation scenario t containing the simulation's parameters.

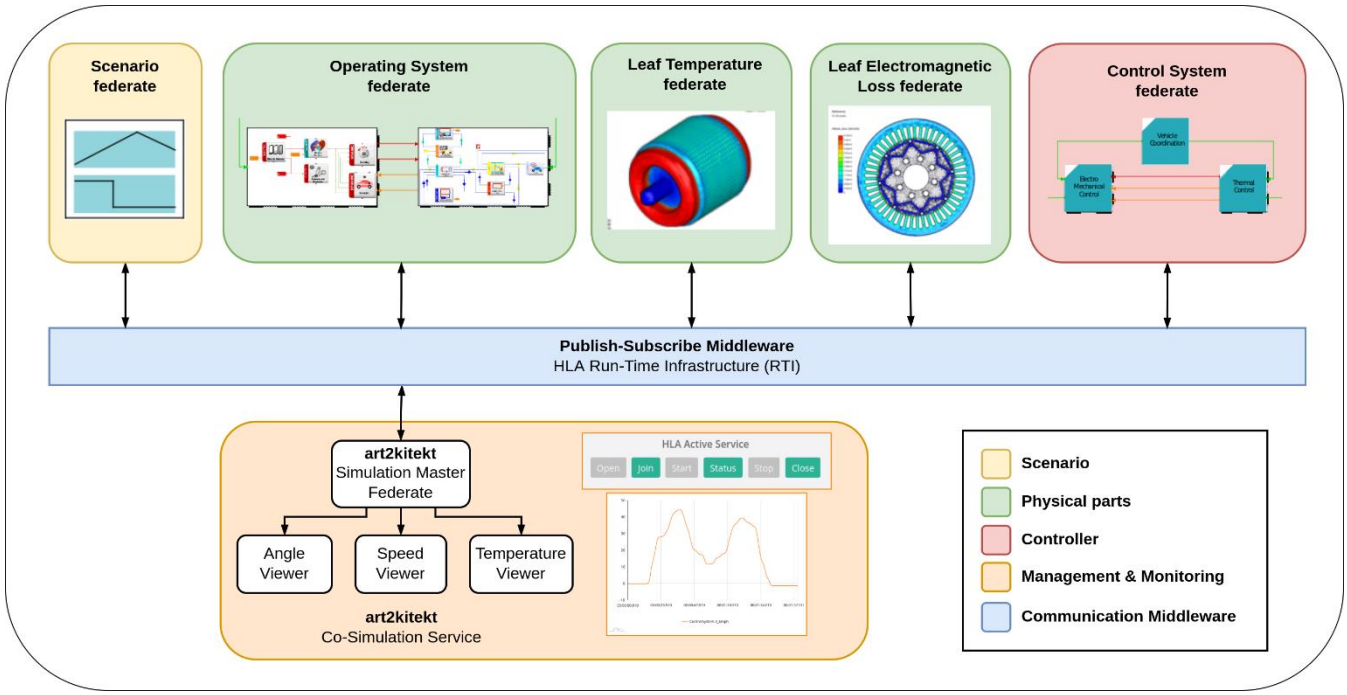


Figure 59: Distributed Co-Simulation Execution

3.1.2 Lessons Learned

Different teams must define and execute a procedure when they work together in a distributed simulation. The cluster members have learned how to do that in the first development step, using a local network. The work demonstrates the viability of the final cluster in a wide network.

Some of the points that the cluster members have learned are:

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

- To obtain benefits using two popular standards: FMI and HLA.
- To specific the connection between the different simulation units.
- To establish simulation time parameters, such as simulation time and step times.
- To design global scenarios and the simulation unit functionalities.
- To set up the simulation parameters.
- To analyse the simulation results with graphs and tables.

This knowledge allows the delivery of the most innovative interoperability solutions. It offers the management, guidance, and technical support that is needed by the industry to the successful integration of distributed simulation systems into interoperable solutions.

To run the cluster in a wide network, we have learned that it is essential to have public servers where A2K can offer simulation services to configure and run the simulations.

The cluster members have also learned that when they write the deliverables, having text editors in the cloud could also be beneficial for managing information efficiently and avoiding incompatibilities.

3.1.3 KPI

Question	Proposed success criterion/criteria	Result
Can we connect several softwares or FMI/FMUs using HLA?	Success = Connection is working and stable	Multiple heterogeneous tools were gathered inside a single simulation **Success**
Have the capability to distribute the softwares and calculatons?	Success = Several softwares and simulations are used seamlessly from different location and platform	Several softwares and simulations were used seamlessly from different location and platform **Success**
Test the constructed process on an industrial scenario	Success = The scenario provided by industrial partners was working from end to end and shows a noticeable increase in setup efficiency for the industrial partners	No industrial partners provided a valid scenario, and the process was tested on a made-up scenario which should represent industrial challenges. No industrial feedback was thus given **Partial Success**

Table 10: heterogeneous co-simulation KPIs

3.2 SCENARIO-BASED SIMULATION

The goal of this cluster was to demonstrate the efficiency of using scenarios as a means for simulation-based verification in multiple domains of cyber-physical systems ranging from shop floor simulation to crane simulation. Scenario Definition Languages can also be used with Domain-Specific Languages (DSL) and monitoring tools. The process of scenario development can be complex and time-consuming. Using simple constructs to build and express ontologies, TUC aims that its SDL will help simplify the scenario development process and make it accessible for different CPS to use scenarios for its safety assessment. The standard XML format supported by the tool allows sharing scenarios among the stakeholders, and with parsing, it can be used with other applications as simulator configuration.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

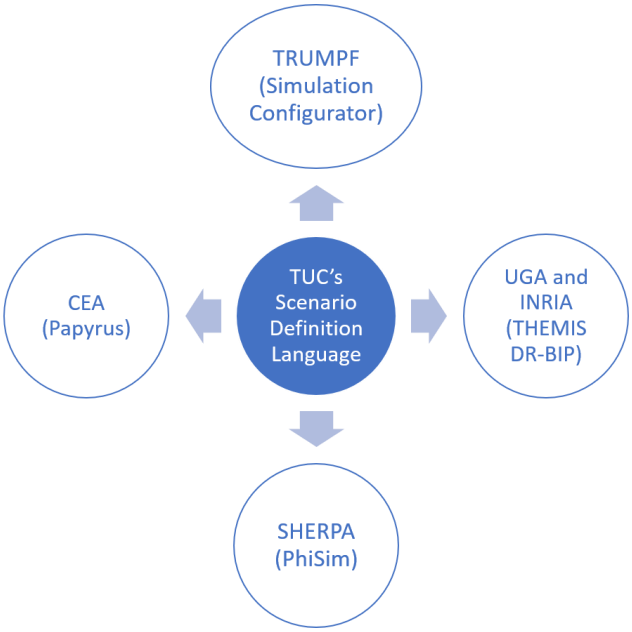


Figure 60 - Scenario based Simulation Cluster Overview

The figure above shows the TUC's scenario language and associated toolset at the cluster's centre, driving the verification scenarios for various partners. This was done by integrating scenarios with a combination of different use cases and simulation platforms. The scenario generation and parsing scripts made it possible to generate varied scenarios. The language was tested using the following use cases and their associated platforms

Partner	CPS Involved	Simulation Environment
CEA	Deep Learning Algorithms for Drones	**Microsoft AirSim**
TRUMPF	Factory floor consisting machines to process sheets	**AnyLogic**
Sherpa	Hybrid Electric Vehicle	**Simulink**
UGA/INRIA	Collaborative Lifting scenario from WP8 with WIKA consisting of Drone and Cranes	**Gazebo (Drone)** **Simulink (Crane)**

3.2.1 Lessons learned

The partners in the scenario simulation cluster coordinated and demonstrated the use of scenarios. These were the key lessons:

- Scenarios can be used for better management of test cases for verification
- Scenario modelling helps in identifying all the elements for requirements that need to be validated and provides structure to the approach.
- The approach still requires some manual work towards the end, with necessary scripts to translate the scenario file to the simulation configuration
- A full-scale CPS system/ subsystem would be an ideal demonstration for the scenario-based approach
- Automatic generation of scenarios from the domain model would amplify the testing process

Note: Please review section 2.6.5 for additional points related to this section.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

3.2.2 KPI

Question	Proposed success criterion/criteria	Result
User scenarios and multiple cyber-physical systems	Success = The tool enables the user to have access to different cyber-physical systems and generate multiple scenarios	**Success** Represents a simplified language that models the important constructs related to the validation required of different types of CPS
The scenario-based approach simplifies the verifications	Success = Simulation-based Verification and debugging were simplified by the methods developed	**Success** The approach combines domain modelling with effective pruning of individual scenarios that allow systemic generation of test cases
Test the constructed process in an industrial scenario	Success = The scenario provided by industrial partners was working from end to end	**Success** The approach was tested on four different CPS use cases

Note: Please review section 2.6.6 for additional points related to this section.

3.3 MODELLING AND ANALYSIS OF AI-BASED SYSTEMS

3.3.1 Summary

In this cluster, we have identified and explored the best available tools interaction for a feasible implementation. Each of the efforts done separately within each tooling environment (section 2) aim to enable seamless interaction of these tools. This interaction has two main purposes: 1) Enable each stakeholder to work in their current environment, maintaining their current best practices, while transparently applying constraints coming from other viewpoints, and 2) Provide formal, standardized and machine-readable representations of the models, so that automatic reasoning and smart applications can be built on top of these models.

To this end, it is necessary to have the capability to exchange formal and standardized models, enhance analysis capabilities via reasoning, and to enhance the tools with knowledge acquisition & reuse capabilities. This is an ambitious and multidisciplinary challenge, which requires the interaction of different approaches, technologies, concerns and viewpoints. Figure 61, shows the refined tool chain, with a further selection of tools for which the interaction is envisaged. In this new configuration, the main tools considered regard Model-Based System Design (Papyrus Modeller), Model Based Safety Analysis (Papyrus Sophia) and Verification (Isaieh, Colibrics/Colibri2). This refinement task was not trivial, since the viewpoints, the levels of abstraction and the domains for each tool have to be aligned.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

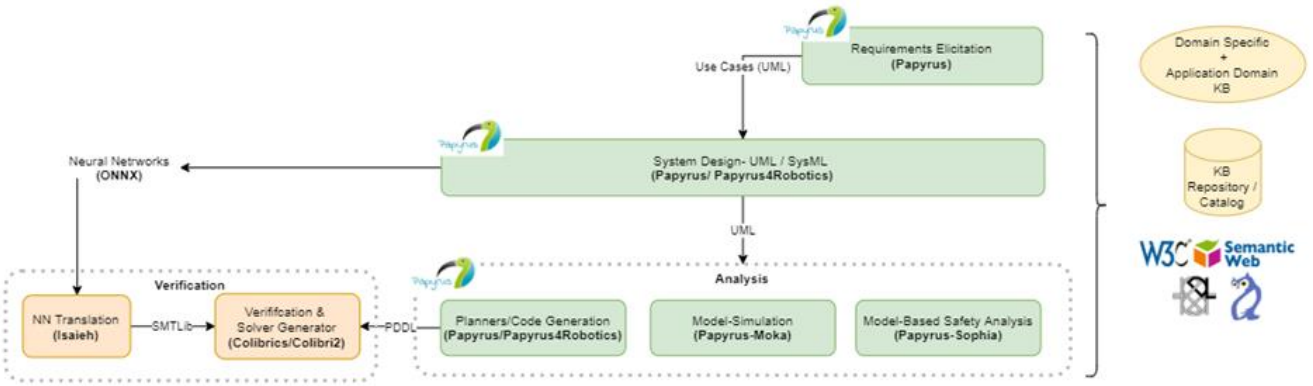


Figure 61: Refined toolchain for Modelling and Analysis of AI-Based Systems cluster

In the following sections, we present the overall achievements and how these contribute to fill the gap to enable interaction among the cluster's tools.

3.3.1.1 MODEL-BASED ENGINEERING and KNOWLEDGE-REPRESENTATION INTEGRATION

Regarding MBSE and Ontologies, we have successfully provided a domain specific standardized language for drones. This standardized ontology, compatible with CORA, targets the physical components in a drone system, their parts and their relations. Thanks to our approach, we can extract UML designs from Papyrus for System Design, as OWL representations, which carry the semantics of ODrone and CORA. We have also shown how this transformation preserves the semantics, and how the expert knowledge from CORA can be evaluated against the annotated model. Thus, we have provided a POC and a demo of the generation of tool-agnostic models of systems, which can be shared in a larger ecosystem, and how external constraints (CORA) can be applied.

In order to deliver trustworthy AI-systems, their capabilities need to be compliant with the constraints imposed by their intended usage. A system can be deployed in different environments and integrated in unforeseen ways with new pieces of technology. To ensure the integration is compatible and that the provided services and behaviour of the system remain within acceptable parameters, it is required to automatically understand these capabilities and their requirements. By integrating knowledge in the engineering process in a standardized manner (knowledge based engineering), we expose the relevant capabilities, properties and parameters, so that other tools can provide their evaluation and analysis to assist and enhance the design process. This is achieved by the annotation of the entities in the UML design with the corresponding recommended domain specific vocabularies (ontologies).

There is at this point no direct (automatic) interaction with Safety and Hazard constraints yet, but having the shareable-annotated models is necessary to achieve this interaction.

3.3.1.2 Integration safety and KBSE

We develop a prototype tool to enable the Hazard identification in AI-based systems based on the Operational Design Domain (ODD) definition of such system. The ODD characterizes the Operating Conditions (OCs) where an AS can operate safely. The OCs are any relevant parameters that describe the system's usage scenarios, including environmental conditions, dynamic elements, and scenery.

Our tool used an ontological model to link the different conditions from the system's environments including external agents, and the systems constituents itself, i.e. the existing Hardware and Software (HW/SW) solutions that implement the system functions. This modelling between OC and system features enables the interpretation and reasoning on the operating scenarios.

In addition, the ontology includes an ontological interpretation of the hazard concept to define an explicit representation of the knowledge of hazards and their relations with the system under analysis and its environment. We derive the hazard and related safety concepts from our safety expertise background (Sophia profile). This further integration enable to formalize the implication of the OCs within the causal chains that may lead to adverse behaviours or accidents. For example, if the system's perception is camera based and if the raining condition is in the operational domain of the system, then the triggering event of water on the camera lens and its possible effect (i.e., an undetected dynamic object and a collision with it) are included in the ontology.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

From an exemplification on a drone system in an urban environment landing scenario, the ontological model will include some environmental conditions as Attribute and State (e.g., “Precipitation”, “Day/Night” respectively) with their properties (e.g., “SnowfallIntenisty”) and metrics (e.g., “Kilometer”). The possible influences of these conditions on risk are depicted through Relator and event concepts (e.g., “Precipitation” Participates on the TriggeringEvent “WaterOnCameraLens”). The model also presents some Agent (e.g., “EgoDrone”, “Pedestrian”) with their relevant properties (e.g., “PedestrianZonePresence”) and possible actions (e.g., “EgoTurning”, “EnterLandingArea”). Some ActionEvent participates in hazardous causal chains leading to a MishapEvent, e.g., action “Egoturning” triggers the malfunction “WrongTurningForce” that lead to “DeviatedTrajectory” and finally to a “CrashOnPedestrian” Mishap. These participations represent misuses or wrong decisions.

We model the relations between OCs, vehicle capacities, possible user actions, and hazards in an ontology as an extension of the Unified Foundational Ontology (UFO) modelling language – defined as an UML profile.

3.3.1.3 Integration scenarios & simulation

The Scenario-Based Simulation cluster provides a tool and a methodology to formally define scenarios thanks to their ODME tool (Operational Domain Modelling Environment). The main components of scenarios include the systems and subsystems of interest (entities), the environmental conditions & the course of events. One of the main areas of work for the partners dealing with scenarios is avionics. On the other hand, at the CEA we have been experimenting with different configurations of a drone system, guided by AI/DNN-based components for perception and control (AI/DNN-based UAV navigation) to constantly define & review the path to be followed to accomplish a mission. The scenario under which this system can perform, it is highly dependent on the weather conditions, since, for example heavy rain will affect the quality of the images detected by the camera. DNNs components are trained with images collected in perfect/ideal environmental conditions that are free of noise or corruptions due to environmental conditions, and thus the neural network controller might be less performant under adverse weather conditions. We have targeted weather conditions and the gates positioning in the scenario, which have been defined and formalized the ODME tool. The tools output (the scenario description) is processed by the CEA Drone-Use case experimentation framework. The framework reads the scenario and talks to AirSim through the simulator API to setup the environmental conditions in the UAV world.

Thus a set of scenarios have been generated by the ODME tool, which have been directly consumed by the simulation experiments in AirSim, to define which scenarios present more difficulties for the drone, and on which of them the system’s performance remains nominal. This not only saves time and makes the scenario testing process more systematic, but it also helps to validate the systems ODDs (Operational Domain Definition). This interaction illustrates the potential of the selected tools interaction.

3.3.2 Lessons learned

Note that, even though these works have been developed in parallel, the interrelation between them is straightforward. Indeed, our safety-oriented ontology integrates the specification of autonomous systems’ components; this is the same information managed through the Drone ontology. Similarly, the conditions that have been used for scenarios generation, is information that must be captured a priori within the ODD of the system. The generated scenarios [as a combination of OCs] represent the situation catalogue needed as input for the Hazard identification.

3.3.3 KPIs

In this section, we present the KPIs at a cluster level, focused on the interaction of the different tools (and respective approaches) for which interaction has been envisaged (Figure 61). The main goal of the cluster is to enable interaction between these tools, and reduce the gap between each domain’s representations (safety, system design, simulation, etc.) by showing the feasibility of the integration and proposing feasible solutions.

Question	Proposed success criterion/criteria	Result
Is the integration of Safety Analysis and System Design	Success= show interaction between both approaches	**Success** Since we have been able to apply Papyrus

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

possible?		Sophia's methodology for safety assessment on UML/SysML models. We have also used ontologies to formalize safety analysis, thus showing the interaction between KBs and safety approaches.
Is the integration of Requirement Elicitation and System Design possible? **Is the integration of Requirement Elicitation and Knowledge engineering possible?**	Success= show interaction between both approaches	**Success** In Kochbati's thesis[1,2] developed at CEA during CPS4EU, we use NLP techniques to translate textual requirements as UML use case models.
Is the integration between Simulation and Safety Analysis possible?	Success= show interaction between both approaches	**Success** Thanks to the scenario generation and its formal background using ontologies, we have feed the simulator (AirSim) with the parameters coming from the scenarios for the Drone use-case[3]. These scenarios have been defined using TUC's tool (ODME).
Is the integration between KBs and System Design possible? Can automatic reasoning be done on these models?	Success= show interaction between both approaches	**Success** This interaction has been possible thanks to the selection of an upper standardized ontology (CORA), the engineering and integration of a Domain Specific Ontology (ODrone) into CORA, and the exploitation of this KB to annotate UML systems models in Papyrus. We have also presented various reasoning tasks[4] on the annotated models, which show the consistency of the integration and how these models can be exploited.
Can knowledge in the Autonomous Systems Domain regarding Systems Design, Scenarios, Safety Constraints and Simulation be formalized using ontologies?	Success = Enable the different tools dedicated to each application domain to represent their content as formal ontologies.	**Success** While targeting the semantic interaction of safety, system design, scenarios definition and simulation, each application domain has established "upper" ontologies to integrate their respective models, and obtain OWL representations (to various extents) of the models in each tool.
Are the formal representations of each domain concerns compatible? Tool-agnostic? Can these	Success = Demonstrate interaction between 2 or more tools.	**Success** A direct interaction of the models generated by all tools has been a guideline but remains a goal out of the scope of CPS4EU.

[1] *Kochbati, T., Li, S., Gérard, S., & Mraidha, C. (2021). From User Stories to Models: A Machine Learning Empowered Automation. In MODELSWARD (pp. 28-40)*

[2] *Takwa Kochbati: Bridging the gap between natural language system requirements and architecture design models. (Combler le fossé entre les exigences du système exprimées en langage naturel et les modèles de conception d'architecture). University of Paris-Saclay, France, 2021*

[3] *Arnez, Fabio, Ansgar Radermacher, and Huascar Espinoza. "Quantifying and Using System Uncertainty in UAV Navigation." arXiv preprint arXiv:2206.01953 (2022).*

[4] *Medinacelli, L. P., Mraidha, C., Noyrit, F., Augmenting Model-Based Systems Engineering with Knowledge, (submission) MDE Intelligence workshop, Models 2022.*

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

models interact?		Nevertheless, to achieve this goal not only the tools need to be able to interact with the KB environment, but also their concerns, viewpoints, etc., have to be aligned. Thus, first, we have enabled semantic technologies to interact with most of the tools directly, and then we have shown the feasibility of the cluster's concept through the main interactions presented previously. The full cluster's interaction remains in a loose integration phase, in part, because not all tools use-cases are sufficiently aligned, nor the formalization is mature enough.

Table 11: Modelling and analysis of AI-based systems

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.

4 CONCLUSION

WP5 was to define and implement pre-integrated architectures as a multi-purpose HW-SW system. To build these architectures, enabling technologies must be boosted and integrated into a supply chain to design, validate, produce, and qualify the CPS. A success criterion to achieve the goal relies on defining a full set of design and validation tools aimed to increase efficiency and productivity. These tools should include design and validation of AI components, modelling, and simulation of the control of CPS, from components to large systems, applications virtualization of specific pre-architecture, tools, methods, and processes ensuring dependability and performance properties of such pre-architectures.

Today's industry is increasingly international, collaborative, multi-actors, and multi-site and so are supply-chains and production operations. Even software modelling and development are spread all over the planet to form more and more globalized companies. A manufacturer needs several tools to cover a task in its engineering activities like modelling, simulation, or testing.

Tools integration expertise is however not the responsibility of the engineer who makes use of these tools. Every engineer must have the ability to build his own association of tools to achieve the task he has been given. To enable this, current technological development allows more and more distributed collaborative work based on high-performance communication networks and emerging concepts such as edge computing or on cloud approaches.

Based on these observations, the members of WP5 proposed to revisit the existing standards, technological developments to provide an answer to those issues raised by European industrial network and make available tools in this multi-actor, distributed, and collaborative framework.

Our approach is to build a distributed infrastructure for design, simulation, and testing based on three pillars that altogether will enable the CPS vision:

- Explore and evaluate the use of the HLA standard in the world of cyber-physics to allow collaborative and distributed work despite different levels of maturity for each of the technologies involved. HLA standard was shown to allow rapid prototyping and agile integration. A demonstrator was set up on a base offered by ITI and Sherpa then enhanced by all the partners.
- To integrate the difference of culture of engineer, an Ontology Driven System Design was used. This concept supports the collaboration between several actors with different cultures and unify their tools usage and requirements. Ontology usage also enabled research on trusted AI carried out by the CEA. It also allowed the workgroup to set up the test campaigns and to execute the simulations, while preserving the semantics defined by each functional team involved in the project. TUC had worked on test generation and evaluating the HLA-based distributed simulation.
- To test an integrated automatic code generation and automated parallelization capability, we made a first study while EMX was participating. Unfortunately, the reduction of EMX contribution to the CPS4EU project put an end to that test, but showed that the state of the art was sufficiently advanced in code generation and parallelization today to support our needs.

 This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276.