AIDOaRt

AI-augmented automation supporting modelling, coding, testing, monitoring and continuous development in Cyber-Physical Systems

D 3.4 - AIDOaRt Core Infrastructure and Framework - Final Version

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007350. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Sweden, Austria, Czech Republic, Finland, France, Italy, Spain. This document reflects only the author's view and that the Commission is not responsible for any use that may be made of the information it contains.

Contract number:	101007350
Project acronym:	AIDOaRt
Project title:	AI-augmented automation supporting modelling, coding, testing, monitoring and continuous development in Cyber-Physical Systems
Delivery Date:	July 31, 2023
Authors:	Hugo Bruneliere (IMTA) – Initial draft + Integration of multiple contents Bilal SAID (SOFT) – Modelio generation support Representatives of all the involved Solution and Use Case providers via their contribution to the project's common Modelio model and their inputs collected using different template documents (cf. below).
Contributing Partners:	ABI, ACO, AIT, AND, AST, IMTA, INT, ITI, JKU, ROTECH, MDU, PRO, SOFT, TEK, TUG, UCAN, UNIVAQ, UOC, VCE
Date:	July 31, 2023
Version	V1
Revision:	08
Abstract:	This deliverable marks the formal ending of WP3. It is a consolidated version of D3.2 and D3.3, taking into account the more up-to-date feedback from the involved Solution Providers and Use Case Providers and integrating the latest developments performed in the context of WP3. This deliverable describes the current (and final) status of the AIDOaRt Core Tool Set that is developed within the context of Tasks T3.2 (Definition and development of the AIDOaRT infrastructure), T3.3 (Design and development of the AIDOaRT framework), and T3.4 (Design and development of the AIDOaRT services). To conclude WP3, it provides the final and up-to-date version of the mapping of the generic components defined in the Core Tool Set to: (1) specific solutions (i.e. tools) and capabilities (provided by Solution Providers), and (2) use case requirements (provided by Use Case Providers) that can be potentially satisfied through those components. This deliverable also provides details on the current development status and plans of the different solutions supporting the AIDOaRt Core Tool Set components, as well as on how they have already been applied in the context of some use case challenges and related cyber-physical systems.
Status:	Type: Other, Dissemination Level: PU

DOCUMENT REVISION LOG

VERSION	REVISION	DATE	DESCRIPTION	AUTHOR
V1	01	02/2023	Initial Draft	Hugo Bruneliere (IMTA), in coordination with Sergio Morales (UOC) for D2.3
	02	03/2023	Detailed Document Structure	Hugo Bruneliere (IMTA)
	03	02/04/2023	First Document Generation	Bilal Said (SOFT)
	04	05/2023 – 06/2023	Collection of content from Modelio and via the different shared template documents.	Hugo Bruneliere (IMTA), Bilal Said (SOFT), All the involved partners providing content via Modelio and template documents
	05	03-05/07/2023	Draft and integration of the different contents in the various sections	Hugo Bruneliere (IMTA)
	06	06/07/2023	Complete draft for Internal Review by: • Ken Sharman (ITI) • Fernando Herrera (ACO) And for External Review by: • Mehrdad Saadatmand (RISE) • Ismael Torres Boigues (PRO)	Hugo Bruneliere (IMTA) + All the involved partners …
	07	27/07/2023	Updated version according to the comments received from the various reviewers	Hugo Bruneliere (IMTA) + All the involved partners …
	08	28/07/2023	Final version for EU Submission	Hugo Bruneliere (IMTA) + All the involved partners …

Executive Summary

The AIDOaRt approach heavily rely on the use of Model Driven Engineering (MDE), combined with Artificial Intelligence (AI) techniques, at different levels of the continuous development (i.e., DevOps) process of large and complex Cyber-Physical Systems (CPSs). This implies dealing with a potentially large number of heterogeneous models that are designed, created and/or handled at both system design time and runtime/execution time (cf. WP2 for the collection of data models). This also implies dealing efficiently with several types of inter-model relationships that may exist to appropriately feed the AI-augmented features supporting different continuous development activities (cf. WP4). Thus, AIDOaRt aims to provide scalable support for managing models, related data, and corresponding generic services. The main goal of WP3 is to offer the proper support for efficiently handling and analysing the different artefacts (e.g., models) resulting from both the continuous development process of the considered system and the data collection process performed thanks to WP2. These artefacts/models are then meant to be consumed and used in the context of the AI-augmented capabilities provided by WP4.

In order to realise this in practice, the AIDOaRt Core Tool Set has been designed in the first phase of the project, cf. the corresponding deliverable [AIDOART-D3.2]. During the second phase of the project, this tool set has then been developed, deployed and experimented in the context of the project use cases [AIDOART-D3.3]. The tool set comprises several solutions mostly coming from our project partners (and possibly completed by a few others according to the needs) that are being combined to achieve the above-mentioned WP3 objectives. As the glue between WP2 and WP4, this Core Tool Set is being developed in strong collaboration with its siblings: AIDOaRt Data Engineering Tool Set and AIDOaRt AI-Augmented Tool Set, cf. the corresponding deliverables for more details on these.

Technically, the Core Tool Set is principally relying on two main environments: Eclipse/EMF and Modelio. Eclipse/EMF is a de-facto standard modelling environment/framework on which many other model-based solutions already rely, including a significant number of solutions in the AIDOaRt project. As an alternative, Modelio is another open-source modelling environment that is quite popular in the industry and used by some partners in the project. Anyway, thanks to such complementary modelling environments, we aim to cover the AIDOaRt use case scenarios as well as possible future scenarios from other companies.

The AIDOaRt Core Tool Set provides complementary capabilities supported by different solutions that can be combined to achieve the targeted goals. Following up deliverable [AIDOART-D3.2] which provided the initial version the Core Tool Set and introduced its baseline characteristics, and deliverable [AIDOART-D3.3] which described an intermediate version of this Core Tool Set, this deliverable D3.4 now provides final details on the latest development status and plans regarding the different solutions supporting the AIDOaRt Core Tool Set components. It also describes how they have already been applied in the context of some recent use case challenges and related cyber-physical systems. Finally, as closing WP3, this deliverable provides the latest version of the mapping of the generic components defined in the Core Tool Set to: (1) specific solutions and capabilities (provided by

Solution Providers), and (2) use case requirements (provided by Use Case Providers) that can be satisfied through those components.

This present deliverable actually concludes the work conducted in the context of WP3. However, the solution providers will continue working on their respective solutions and collaborating with the use case providers on their various challenges and scenarios. As a consequence, the consolidation and practical applications of the various solutions from the AIDOaRt Core Tool Set will be continued in the context of both WP4 (for the AI-related aspects of these solutions and the support they provide for AI-augmentation) and WP5 (for the integration of these solutions in the overall AIDOaRt framework and in the different use cases) until the end of the project, and likely afterwards.

Table of Contents

Key Terminology Abbreviations

Abbreviations	Terminology
AI	Artificial Intelligence
AIOps	AI Operations
ANN	Artificial Neural Network
API	Application Programming Interface
BPMN	Business Process Modelling Notation
CPS	Cyber-Physical Systems
CPSoS	Cyber-Physical Systems of Systems
DevOps	Development Operations
DSL	Domain-Specific Language
EMF	Eclipse Modelling Framework
IoT	Internet of Things
JSON	JavaScript Object Notation
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GAN	Generative Adversarial Network
MARTE	Modelling and Analysis of Real-Time and Embedded systems
KPI	Key Performance Indicator
MBE	Model-Based Engineering
MBRE	Model-Based Requirements Engineering
MDE	Model-Driven Engineering
ML	Machine Learning
M2M	Model-to-Model transformation
M2T	Model-to-Text transformation
REST	Representational State Transfer
SE	Systems and Software Engineering
SLR	Systematic Literature Review
SMS	Systematic Mapping Study
SysML	Systems Modelling Language
TMS	Thermal Management System
UML	Unified Modelling Language
WP	Work Package
XMI	XML Metadata Interchange
XML	Extensible Markup Language
YAML	YAML Ain't Markup Language

Partners Names Acronyms

In the following table, the partners acronyms and their full names are listed. The short acronyms are used throughout the deliverable text to identify the partner providing a particular case study requirement or data requirement, or providing a given solution.

Partner Name Acronym	Full Partner Name
ABI	Abinsula SRL
ABO	Åbo Akademi
ACO	ACORDE Technologies S.A.
AIT	AIT Austrian Institute of Technology GmbH
AND	Anders Innovations Oy
AST	Automated Software Testing GmbH
AVL	AVL List GmbH
BT	Bombardier Transportation
CAMEA	CAMEA, spol. s r.o.
CSY	CLEARSY SAS
DT	Dynatrace Austria GmbH
HIB	HI Iberia Ingeniería y Proyectos S.L.
IMTA	Institut Mines-Telecom Atlantique Bretagne-Pays de la Loire
INT	Intecs Solutions S.p.A.
ITI	Instituto Tecnológico de Informática
JKU	Johannes Kepler University Linz
MDH	Maelardalens Hoegskola (Coordinator)
PIO	Previsionio
PRO	Prodevelop SL
QEN	Qentinel Oy
RISE	RISE Research Institutes of Sweden
ROTECH	Ro Technology srl
SOFT	Softeam
TEK	Tekne SRL
TUG	Technische Universitaet Graz
UCAN	Universidad de Cantabria
UNISS	Università degli Studi di Sassari
UNIVAQ	Università degli Studi dell'Aquila
UOC	Fundació per a la Universitat Oberta de Catalunya
VCE	Volvo Construction Equipment AB
WESTMO	Westermo Network Technologies AB

1 Introduction

As described in the AIDOaRt project proposal and already introduced in deliverable [AIDOART-D3.1] – "Report on Foundations of MDE and AIOPS for DevOps", AIDOaRt aims at combining Model Driven Engineering (MDE) and Artificial Intelligence (AI) principles and techniques to improve the support for the continuous development (a.k.a. DevOps process) of large and complex Cyber-Physical Systems (CPSs). More particularly, the objective of WP3 is to elaborate on the required capabilities that allow the efficient handling of the various kinds of data artefacts (mostly data models) obtained in WP2, along with the other kinds of available software and system engineering models. Thanks to WP3 capabilities, these interconnected models can then be used to feed the AI-augmented features provided in WP4. As a result of the work in WP3, the AIDOaRt Core Tool Set has been designed in deliverable [AIDOART-D3.2] and has been further developed in deliverable [AIDOART-D3.3] to provide the AIDOaRt core infrastructure and "framework". Moreover, the AIDOaRt Core Tool Set has also been experimented with in the context of various project use cases (cf. the work in WP1 and WP5).

In deliverable [AIDOART-D3.1], we started by studying the state-of-the-art of corresponding scientific and technical aspects to be possibly considered, reused, extended, and/or refined in the context of WP3 (but also in the context of the other technical work packages, i.e., WP2 and WP4). This effort concerned existing approaches at the intersection of two or three of the AIDOaRt main areas, namely MDE, AI (and more specifically Machine Learning) and DevOps.

In deliverable [AIDOART-D1.1], the use case partners provided their initial general needs for the AIDOaRt solution and detailed further their use case scenarios as well as underlying CPSs according to these needs. In deliverable [AIDOART-D1.2] and then in deliverable [AIDOART-D1.4], we defined the architecture of the overall AIDOaRt solution, collecting inputs on individual solutions from the solution providers in the project.

Based on the work performed and reported in the already released deliverables, deliverable [AIDOART-D3.2] provided a realistic vision of the AIDOaRt Core Tool Set. We identified an initial list of technical solutions developed and/or provided by the project partners to support the AIDOaRt Core Tool Set components. Based on the work realised in the context of WP1, we also connected the various use case requirements to these components in order to assess the situation in terms of possible coverage of the use case providers' needs by the solution providers.

As a follow-up, in deliverable [AIDOART-D3.2], we went one step further and provided more details on the intermediate development status of the different solutions supporting the AIDOaRt Core Tool Set components, as well as on how they were already applied in practice at mid-project in the context of some use case challenges and related CPSs. We also provided a lower-level version of the mapping of the generic components defined in the Core Tool Set to: (1) specific solutions and capabilities (provided by Solution Providers), and (2) use case requirements (provided by Use Case Providers) that can be satisfied through those components.

With this deliverable D3.4, we formally conclude the work in the context of WP3. To this end, we provides details on the latest development status and plans regarding the different solutions supporting the AIDOaRt Core Tool Set components. We also describes how these solutions have been once again applied in practice in the context of some recent use case challenges and related cyber-physical systems. Finally, we present the latest and up-to-date version of the abovementioned mapping.

The document is structured as follows. Section 2 details the subset of the AIDOaRt Core Tool Set solutions having capabilities released or to be released between M20 and the end of the project, explains how their development has progressed until now and the future plans regarding them. Then, Section 3 presents the latest up-to-date version of the mapping between all the solutions and the components of the AIDOaRt Core Tool Set, at the interface-level. Similarly, Section 4 presents the latest up-to-date version of the mapping between the use case requirements and the components of the AIDOaRt Core Tool Set, also at the interface-level. Section 5 describes recent and ongoing applications of solutions from the AIDOaRt Core Tool Set in the context of various use case challenges and related cyber-physical systems, while Section 6 concludes the deliverable.

2 Progress status of the AIDOaRt Core Tool Set

This section gives an overview of the implementation achievements, development status, and future plans of the AIDOaRt Core Tool Set solutions with regards to the Intermediate milestone, as described in the previous deliverables [AIDOART-D3.2] and [AIDOART-D3.3]. The past and current achievements, as well as future development plans, of each solution are presented and commented in this section, in order to make explicit the concrete realisations in terms of capabilities development while we are formally concluding the work in the context of WP3.

For each of the concerned solutions, the corresponding solution provider presents:

- In the first subsection "Capabilities Implementation Status", the features released from M21 to M28 (i.e., the submission date of deliverable D3.4), thus covering features with release dates declared at MS5 (M24) and MS6 (M28) of the project.
- In the second subsection "Future Capabilities Implementation Roadmap", the features to be released from M29 up to M36 (i.e., after the submission date of deliverable D3.4 and up to the end of the project), thus covering features with release dates declared at MS7 (M32) and MS8 (M36) of the project.

To briefly summarise the reported status, it should be noted that:

- 28 individual tools are associated to the AIDOaRt Core Tool Set in total;
- Among them, 20 tools are directly concerned by this final deliverable D3.4;
- The past work on the other 8 tools was reported in the previous deliverable [AIDOART-D3.3].

2.1 Solution - ESDE (ACO)

ESDE is an ESL embedded Software Development Environment that is being developed by ACORDE to improve embedded software design productivity (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Explainability (Generic Services)
- Model-Based Capabilities (Data and Modelling)

2.1.1 News and Updates

Figure 1 sums up the status of the ESDE extension produced in AIDOaRt at the time of this report writing. The figure provides a global view and the relation with the AIDOaRt architecture (and thus its related technical WPs). The extensions related to data collection and data management services, are related to WP2 activities and explained in Section 1.1 of D2.3. Those extensions serve to collect traces from the virtual embedded system simulation and store them in a time series database. However, all those services, as well as those AI-based analysis services (pink box), would be impossible without the required supporting infrastructure. Such infrastructure is exactly the topic of WP3. The green box reflects the core infrastructure and services concerned by WP3 and required by the ESDE extension targeted in AIDOaRt.

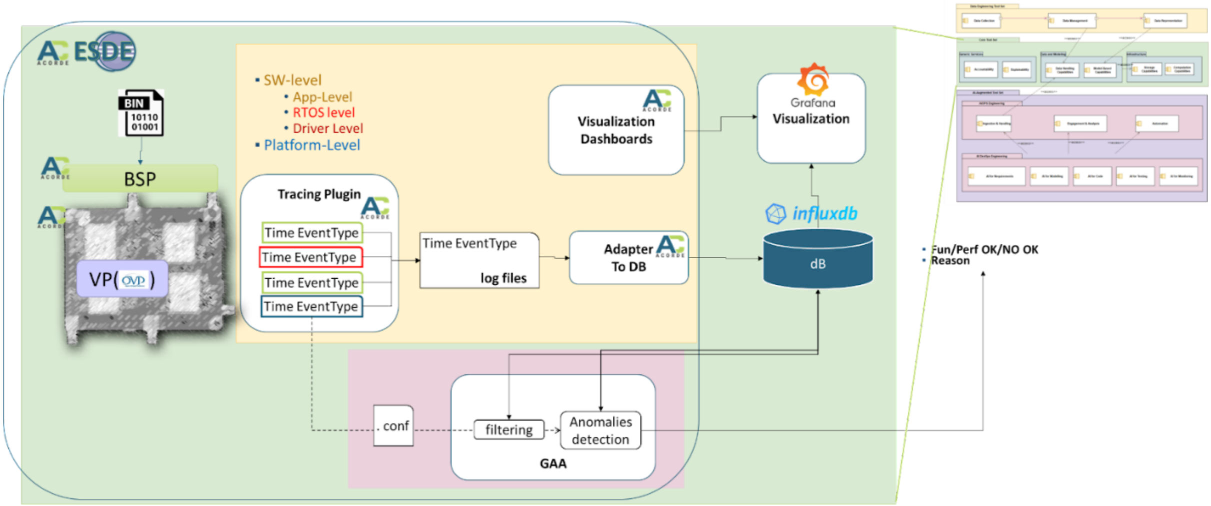

Figure 1 – ESDE multi-level tracing extension relies on an infrastructure setup relying on advanced software componentization and database.

As shown, such a green box has been reflected as a base or environment required by both data collection and analysis services. This infrastructure is such that it can be set up on top of desktop computer (e.g. i5 16GRAM 1t SSD) on ACORDE premises. Specifically, the core infrastructure put in place for the ESDE extended version developed in AIDOaRt is summarised as follows:

- ESDE base framework (AIDOaRt baseline): which accounts for the SystemC library, SC2ES library, Imperas OVP mono-processors environment, EIGEN, GeographicLib, and ACO GNGSS libraries.
- Docker environment working on WSL2 and configuration files for automatic set up of the following services:
    - Time Series Database: Influxdb
    - Visualisation service: Grafana

The last elements, complementing the ESDE baseline for AIDOaRt, have been put in place in the last period. No big update has been done to the system-level part of the framework, i.e. regarding the modelling and software generation capabilities. However, further updates might be expected (in the WP5 context) if further SystemC tests are addressed.

In addition, infrastructure related to AI/ML analysis has to be added. First, a base docker component with Keras has been configured, on top of which the specific deep-learning (DL) solution developed by ACORDE is run. Moreover, some Grafana plugins for visualisation of anomalies intervals have been developed. In any case, the facilities for AI/ML analysis (in purple) are being reported in the context of WP4 (last report was done in section 3.1 of D4.2).

Regarding explainability, ACORDE is currently focused on the interpretation of which sub-set of monitored data is involved in each of the detected anomalies intervals. The work is currently focused on real-time data on the distributed domain (port data in the context of the port monitoring solution), to later focus on embedded system traces from ESDE). The completion of this evaluation will let ACORDE assess the possibility to support explainability as a generic service, i.e. integrated as a part or a plugin of the Generic Anomalies Analysis (GAA) service developed by ACORDE. Therefore, this will require further evolution of ACORDE activity on WP4 and WP5 as planned. Such progress and assessments will be reported in further WP4 (D4.3, D4.4) and WP5 reports.

2.1.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
Automated (or semi-automated) bug detection/prediction: Ability to learn and detect or predict possible functional or performance bugs based on performance traces. Two possibilities envisioned. First one, based on offline trace analysis, can handle non-causal analysis for better anomalies/errors detection. This feature will support product fixes and development smoothly integrated in a DevOps environment. A second possibility is to simulate the system equipped with monitoring probes and a ML engine capable to use the collected metrics for an on-the-fly (real-time, or at least causal) computation for detecting/predicting anomalies.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS5 (M24) **Licence:** Proprietary – ACORDE. (It refers to specific extensions performed on ESDE to detect and/or predict functional and performance bugs developed by ACORDE and considered key for protecting competitive advantages achieved thanks to AIDOaRt. It excludes any open-source and/or third-party source used, and any open-source extension that needs to be published. Eventually, ACORDE might release as open-source fixes and extensions which do not compromise ACORDE competitive advantages granted by AIDOaRt).	Required components, like former prototypes of multi-level tracing and database managing, and a former version of generic anomalies analysis module are in place. The WP3 related work, putting in place specifically suited database and visualisation frameworks complete all the framework components. Further evaluation of this framework is on-going to consider its suitability for the embedded domain. The delivery is replanned by M36, so that use case related data can be considered too.

Table 1 Capabilities Implementation Status of the ESDE Solution

2.2 Solution - DTsynth (AIT)

DTsynth is solution targeting the synthesis of Digital Twins from existing CPS. It is notably based on the use of ML approaches and related techniques (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Explainability (Generic Services)

2.2.1 News and Updates

AIT together with TUG and AVL are pursuing the learning of Digital Twins of cyber physical systems. In this particular context, the learning process of finite state machines from available system data has already been achieved. However, the parameter space reduction in order to speed up the generation of ADAS/AV test case is still being investigated in collaboration with AVL.

2.2.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
Digital Twin Learning: Derivation of a Digital Twin using passive and/or active learning, with a focus on understandable and trustable twins (probably by means of explainable AI and combination with automata learning approaches from TUG)	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS6 (M28) **Licence:** GPL	This has been partially achieved by a collaboration with TUG. However, this feature has already been implemented and tested on a conceptual basis (research prototype).
Digital Twin Learning-Data Derivation: Derivation of stimuli to the original system to get more detailed data to improve an already existing, preliminary Digital Twin. Possibly applying model-based testing techniques.	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS6 (M28) **Licence:** GPL	We are currently evaluating how to integrate the parameter space reduction and other model-based techniques to verify whether a learned model fits the specification using one of our tools. This evaluation work will continue in the coming months.

Table 2 Capabilities Implementation Status of the DTsynth Solution

2.3 Solution - Kolga (AND)

Kólga is a CI/CD tool that helps with building build, test, and deployment pipelines using already existing CI platforms such as GitHub Actions, GitLab CI and Azure Pipelines (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Computation Capabilities (Infrastructure)

2.3.1 News and Updates

An initial demo application has been deployed, and a solution framework with all the components exists and has been made available to the project's partners.

2.3.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
Kólga: Tool for creating CI/CD pipelines using GitLab/GitHub/Azure that can build, test and deliver applications to Kubernetes	**Implementation Level:** Partially Implemented	The codebase from POC level to MVP level still has to be made more robust. The plan is to work on a production ready

clusters. Highly flexible solutions can be created on top of the tool, such as AI testing all using the tools such as Github Actions that could already be in use. An example would be to bring a web service from just source code to running in a scalable production environment in a very short (minutes) time. Our only requirements are that the application is based on a container technology such as Docker and that the final running environment is Kubernetes. In other words, Kólga can also enable running your application in the edge for instance.	**Estimated Delivery Date:** MS7 (M32) **Licence:** MIT	deployment and continuous training / deployment of the application.

Table 3 Future Capabilities Implementation Roadmap of the Kolga Solution

2.4 Solution - Cloud Expertise (AND)

Cloud Expertise provides solutions for cloud architecture, infrastructure, development, etc. via engineers skilled in deployments of most things cloud-related (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Computation Capabilities (Infrastructure)
- Data Handling Capabilities (Data and Modelling)

2.4.1 News and Updates

We studied the use of big query for data lake and, after evaluation, we opted for using it in the context of our solution. Consequently, we deployed corresponding data collectors and data handling demo apps to Google Cloud thanks to our Cloud Expertise.

2.4.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
Cloud expertise: AND is able to provide help and solutions when it comes to Cloud architecture, infrastructure, development and more. We have skilled engineers that can help out with deployments of most things cloud related.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** To be decided based on work done	Data collection capabilities still has to be made more robust in order to deploy cloud ML training / data handling instances. To this end, the plan is to work on improving the collected data formats.

Table 4 Future Capabilities Implementation Roadmap of the Cloud expertise Solution

2.5 Solution - Infrastructure as Code (IaC) Expertise (AND)

AND can provide IaC solutions for things like Cloud infrastructure, Kubernetes setups, database setups, CI/CD providers, and other services (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Storage Capabilities (Infrastructure)

2.5.1 News and Updates

We studied the use of big query for data lake and, after evaluation, we opted for using it in the context of our solution. Consequently, we deployed corresponding data collectors and data handling demo apps to Google Cloud thanks to our IaC Expertise.

2.5.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
IaC expertise: AND is able to provide IaC solutions for things like Cloud infrastructure, Kubernetes setups, database setups, CI/CD providers and other services. We have provided these services for several customers and are confident in our capabilities to provide a robust solution for the main cloud providers.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** To be decided based on work done	We plan to continue this work by studying the performance of big query vs. other relational database management systems regarding collected data.

Table 5 Future Capabilities Implementation Roadmap of the Infrastructure as Code (IaC) expertise Solution

2.6 Solution - devmate (AST)

Devmate is a software suite for automated test case and test code generation, by using model-based and AI-based approaches for test automation and test data determination. (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)
- Data Handling Capabilities (Data and Modelling)
- Storage Capabilities (Infrastructure)

2.6.1 News and Updates

Based on first feedback from interested partners and persons that evaluated the devmate prototype, we are currently working on an enhanced user interface concept for a canvas oriented and graphical model-based test management approach. The Testmodel Editor has had a major update and redesign in the newest version. This makes also the data handling in devmate more easy. The storage capabilities remain unchanged.

2.6.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
Code Parser: Parse input-/output-parameters from existing code or unit-/system-models (e.g. XML / DSL specified models)	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS5 (M24) **Licence:** Proprietary Licence (Automated Software Testing GmbH)	Code Parser already supports C# and Java. C parser is currently in development. A first version is ready for testing, these tests will be performed in the coming months of the project. The produced intermediate model is saved locally.
Testcase Generator: Generating a set of test cases based on rules (data-driven, combination of input parameters / data)	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS6 (M28) **Licence:** Proprietary Licence (Automated Software Testing GmbH)	Testcase Generator is implemented for the most part, with a planned review in the future months. TestGen produces a test-model of the parsed intermediate model, also saved locally on the machine.

Table 6 Capabilities Implementation Status of the devmate Solution

2.6.3 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
Testmodel Editor: Automatically combining and generating test cases for system/unit under test (primarily black-box validation test techniques)	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** Proprietary Licence (Automated Software Testing GmbH)	The updated Testmodel Editor directly manipulates the internal test model. Future plans involve improving usability and providing features to more efficiently manipulate the test model.
Testcase Evaluation: AI/ML augmented evaluation and reduction of test cases (based on an extendable set of metrics e.g. coverage, mutation score, test runtime)	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** Proprietary Licence (Automated Software Testing GmbH)	Future development efforts will revisit metrics and code coverage while current efforts are focused on language support and usability.
Testdata prediction system: AI/ML supported test data specification (test data management, similarity measures, test data prediction system)	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS8 (M36) **Licence:** Proprietary Licence (Automated Software Testing GmbH)	Research into similarity and prediction systems are currently on halt. We are not sure yet whether this feature will actually be treated or not.

Model Parser: Parse input-/output-parameters from existing unit-/system-models (e.g. XML / DSL specified models)	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS8 (M36) **Licence:** Proprietary Licence (Automated Software Testing GmbH)	Models can be already parsed through Code Parser and a more dedicated support will be added on demand.

Table 7 Future Capabilities Implementation Roadmap of the devmate Solution

2.7 Solution - EMF Views (IMTA)

EMF Views is an Eclipse-based solution that brings the concept of database views to the modelling world. It allows creating, navigating, and querying views over model(s) that conform to different metamodels and languages (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)
- Data Handling Capabilities (Data and Modelling)

2.7.1 News and Updates

As mentioned in previous deliverable D3.3, the Viewpoint Builder and View Builder (i.e., the two EMF Views core components) already provide a basic support for Verification and Validation (V&V) based on metamodel conformance, and the associated VPDL language also comes with base syntactic validation. At this stage, we did not yet have faced the need for more advanced V&V capabilities but this could be envisioned in the future (cf. Section 2.7.3). Concerning the scalability and efficiency of model view computation, navigation and querying, several identified bugs have been fixed during this second year of the project. For example, we made the EMF Views solution more reliable when considering UML/SysML models within the specified views, as required notably in the CPS context of the VCE partner (cf. Section 2.7.2). Moreover, as explained in previous deliverable D3.3, the generated views are partially editable by default (for basic attribute modification) but the provided view update capabilities remain limited. During this second year of the project, we started to work on integrating the use of Machine Learning techniques with EMF Views in order to improve the semi-automated computation and/or update of some elements of the views. We plan to achieve more complete and publishable results during the third and last year of the project (cf. Section 2.7.3). This could be useful in practice in a CPS model-based engineering context, for example to recommend eventually missing inter-model relations that the system engineers have not or cannot foresee. We currently plan to report on this latest work in a future WP4 deliverable.

2.7.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes

Model Viewpoint and View computation: Be able to compute a given view in a scalable way, based on a previously specified viewpoint and a corresponding set of metamodels and models.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS6 (M28) **Licence:** EPL 2.0 / GPL 3.0	As introduced earlier, this tool capability is made available in the current version of EMF Views. Updates have been performed (as some bugs have been detected on given view operations) in order to better support the building of viewpoints and views over particular types of models (e.g., UML/SysML ones). For the last phase of the project, we envision extending/refining EMF Views and the associated VPDL language with some AI-related support (cf. the next section / table).
Model Viewpoint and View navigation and query: Be able to efficiently navigate and query an already computed view.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS5 (M24) **Licence:** EPL 2.0 / GPL 3.0	As introduced earlier, this tool capability is made available in the current version of EMF Views. Updates have been performed (as some bugs have been detected on given view operations) in order to better support the use of viewpoints and views over particular types of models (e.g., UML/SysML ones). For the last phase of the project, we envision extending/refining EMF Views and the associated VPDL language with some AI-related support (cf. the next section / table).

Table 8 Capabilities Implementation Status of the EMF Views Solution

2.7.3 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
Model Viewpoint and View update: Be able to dynamically and efficiently update an already computed view.	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** EPL 2.0 / GPL 3.0	As introduced earlier, this tool capability is not yet available in the current version of EMF Views. The work started on integrating the use of Machine Learning techniques with EMF Views in order to improve the semi-automated computation and/or update of some elements of the views. This work will be completed and hopefully published during the third and last year of the project.

	The work on more advanced V&V capabilities over views could be envisioned during the last months of the project if required in the context of a use case (e.g., the VCE one), for instance to facilitate the verification and/or validation of some CPS-related properties.

Table 9 Future Capabilities Implementation Roadmap of the EMF Views Solution

2.8 Solution - INT-XAI (INT)

INT-XAI is a solution that brings the concept of explainability to the deep learning modelling world, thus providing an additional validation method. (cf. the deliverable [AIDOART-D3.3] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Explainability (Generic Services)

2.8.1 News and Updates

As introduced in the previous deliverable D3.3, INT-XAI is designed to assist ML engineers during their modelling development by giving a human interpretable reason for the model output. The artificial intelligence model that provides the testing ground is the one underlying the INT-DET solution: it is an object recognition network based on convolutional networks. Among the latest techniques suitable for this type of model, special attention has been paid to Class Activation Mapping (CAM): CAM is a widely used technique that highlights the discriminative regions of an image by generating a heat map. It uses the class-specific information obtained from the final convolutional layer of the network to localise important regions.

2.8.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
Explainable AI: Complex ML models such as deep neural networks become more suitable for formal analysis when they can be more interpretable/explainable. Explainable AI (XAI) techniques produce "visual explanations" for decisions from a large class of CNN-based models, making them more transparent.	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS6 (M28) **Licence:** Proprietary	Various CAM implementations are currently being studied and tested. A special attention is being given to the development of a solution suitable for operating in complex scenes such as the driving use case. This work will continue in the third year of the project.

Table 10 Capabilities Implementation Status of the INT-XAI Solution

2.9 Solution - a2k-modev (ITI)

The a2k-modev solution provides analysis and simulation facilities for computation and display of various system performance metrics (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)
- Computation Capabilities (Infrastructure)

2.9.1 News and Updates

During the 4th plenary hackathon, we presented the latest updates showing how we can import text-based specifications of the system architecture in the form of simple spreadsheets. This has allowed us to easily create models within the A2K software of both the system hardware and the application software. We also showed first steps in the development of a tool for the simulation of the activities of the smart port use case in terms of what happens when a new ship arrives at the port for loading or unloading.

2.9.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
a2k/scheduling: A scheduling simulation service able to generate synthetic execution traces and to visualise and interpret activity interactions and resource usage.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS5 (M24) **Licence:** Proprietary ITI	This service is already complete and is in use. However, it may be extended in the future with more advanced facilities such as hierarchical scheduling.
a2k/monitoring: A monitoring service to extract temporal and functional execution information from real time systems.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS5 (M24) **Licence:** Proprietary ITI	A basic run-time monitoring service has been implemented. We use this to collect the execution times of tasks. We may update it if needed in the coming year.
a2k/analyser: Probabilistic timing & feasibility analysis of heterogeneous, distributed, real-time systems. Sensitivity analysis. Design space exploration and optimisation.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS5 (M24) **Licence:** Proprietary ITI	We have updated the a2k/analyser service with a new algorithm for analysis of task response times in a distributed CPS. This algorithm uses the method of computational performance analysis to compute upper bounds for the response times. Further updates could be made if needed in the coming year.

Table 11 Capabilities Implementation Status of the a2k-modev Solution

2.10 Solution - a2k-runman (ITI)

The a2k-runman (A2K Run Time Manager) solution aims to monitor the operation of a CPS in real-time and to provide warnings and advice when critical situations are observed or predicted. (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Data Handling Capabilities (Data and Modelling)

2.10.1 News and Updates

In the a2k-runman solution component, we have made significant advances in the a2k/tuning service where we have developed and evaluated some novel prediction algorithms concerning the workload of processing nodes. The idea is to monitor the execution times of running tasks to use machine learning algorithms to predict future workloads. The learning algorithms are trained using synthetic data generated by the a2k/modev and a2k/scheduling services. With these predictions, we then employ a control algorithm to adjust the processor clock frequency in real-time. This in turn controls the power consumption of the processor hardware. The goal is to try and minimise the power consumption. This will be especially useful for the processors (possibly mobile) connected to the IoT devices at the edge of the network.

2.10.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
a2k/tunning: Automatic criticality level / operational mode change management based on real time data collection and analysis using adaptive learning algorithms	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** Proprietary ITI	This component will adjust the processor frequency considering the predicted workload demands. We will use a real-time simulator and a prediction method based on AI to do this. Currently, we are evaluating different CPU frequency control algorithms and how to interface these with the workload prediction methods which are also currently under development. This work will be continued in the coming year.

Table 12 Future Capabilities Implementation Roadmap of the a2k-runman Solution

2.11 Solution - a2k-depman (ITI)

The a2k-depman (A2K Deployment Manager) solution aims to provide information regarding the allocation of computational and communications resources to the various CPS activities at run-time (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)

2.11.1 News and Updates

A new service, a2k/optimiser, was presented at the 4th plenary hackathon. The goal of this service is to automate the deployment of software services to the various processing nodes of a distributed cyber physical system. We are focussing on the use of evolutionary methods (e.g., genetic algorithms) to solve the multi-objective problem of the allocation of services to nodes. We have developed several performance metrics that a2k/optimiser attempts to maximise or minimise.

2.11.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
a2k/optimiser: Deployment assistance in heterogeneous distributed real-time systems using multi-objective optimisation and machine learning	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** Proprietary ITI	This service is still currently under development. We have already built an optimisation framework in MATLAB and are using this to investigate and evaluate different multi-objective optimisation algorithms in the context of dynamic service deployment in the smart port use case. In the future, once suitable optimisation algorithms have been identified, we will port the algorithms to be more integrated within our A2K software.

Table 13 Future Capabilities Implementation Roadmap of the a2k-depman Solution

2.12 Solution - MOMOT (JKU)

MOMoT is a search-based model transformation framework, now supporting not only global searchers for model transformation orchestrations but also local ones. (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)
- Computation Capabilities (Infrastructure)

2.12.1 News and Updates

MOMoT is an already working tool. However, an effort is typically required to configure MOMoT for a particular model-based optimization problem. The main activities consist in i) providing a metamodel to define a domain specific problem and solution models, ii) define in-place model transformation rules that, given a (problem) model conforming to the given metamodel, is able to generate (solution) models that optimise a given (multi-objective) function, iii) the Java code that implement the optimization function leveraging model’s content. JKU is currently using MOMoT “as is” to solve optimization problems for thermal management systems in the context of the VCE use case.

2.12.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
AI-augmented MOMOT: MOMoT is a search-based model transformation framework, now supporting not only global searchers for model transformation orchestrations but also local ones. Thus, it allows multi-objective local and global search.	**Implementation Level:** Partially Fulfilled **Estimated Delivery Date:** MS7 (M32) **Licence:** Eclipse Public Licence - V2.0	At this stage of the project, future AI-augmented extensions are still subject to specific concerns that may arise from adopting MOMoT "as is" in different use cases (e.g., the support in setting up a proper MOMoT configuration).

Table 14 Future Capabilities Implementation Roadmap of the MOMOT Solution

2.13 Solution - Modelling Process Mining Tool (JKU)

The Modelling Process Mining Tool (MPMT) is a research tool which includes two main sub-components: the Modelling Event Recorder (MER) and the open-source PROM tool (cf. the deliverable [AIDOART-D3.3] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)

2.13.1 News and Updates

As introduced before, MPMT includes two main sub-components: MER and PROM. MER is an Eclipse plugin that interacts with Sirius-based graphical editors for EMF-based models, and records users' modelling event logs. Event logs are encoded in the IEEE Standard for eXtensible Event Stream (XES), which provides an XML schema for logs encoding. The MER tool also provides an XES metamodel for encoding event logs as EMF-based models. Figure 2 shows the workflow in MER on a high level. The user starts interacting with the modelling editor (step 1). Then MER attaches itself to the created editing session (step 2) and listens to changes in the EMF-based models being modified within the graphical editors. As the user interacts with the editor and creates, modifies, and deletes graphical elements from the modelling canvas, the MER plugin captures all of the corresponding events. This change observation is done via the EMF Notification API, a built-in API in EMF. In step 3, as the editing session is closed, the plugin stores the collected traces in an XES log file.

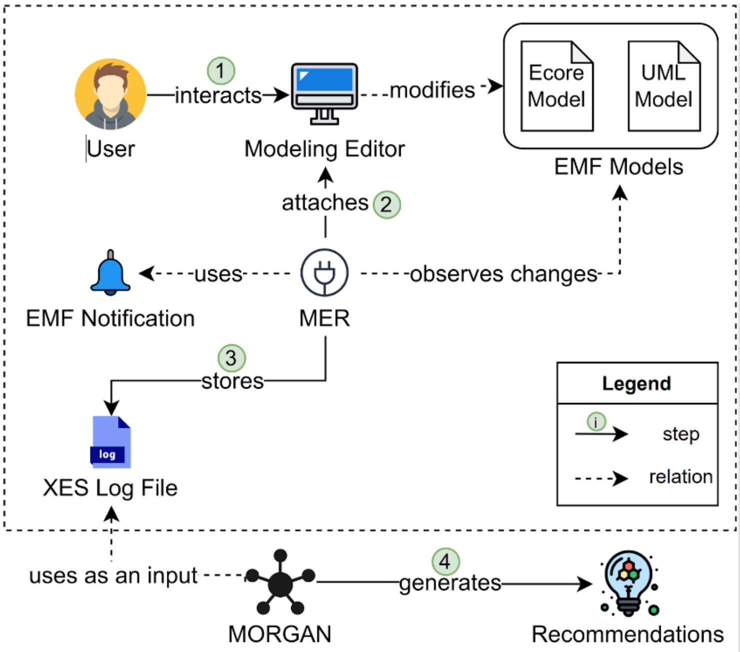

Figure 2 – The MER tool and its integration with MORGAN tool to enable modelling recommendations.

The MER tool is actively used in the challenges proposed by the VCE use case. We are working on integrating the MER tool, to support recording of SysML and AutomationML modelling activities, with the MORGAN recommender tool from UNIVAQ (step 4) to enable modelling recommendations.

2.13.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
Modelling Process Mining: This operation works alongside the Sirius editor plugin and returns an XES log file containing user interaction activities recorded during the modelling phase.	**Implementation Level:** Implemented **Estimated Delivery Date:** MS5 (M24) **Licence:** Eclipse Public Licence - V2.0	The capability is fully working. We are testing the implementation in the context of the VCE use case. The current version is working and able to collect traces while modelling in AutomationML and SysML.

Table 15 Capabilities Implementation Status of the Modelling Process Mining Tool Solution

2.14 Solution - GAN-Based Instance Model Generator (JKU)

The GAN-based Instance Model Generator is a research solution to address data augmentation needs in a Model Driven Engineering (MDE) context. This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)

2.14.1 News and Updates

The GAN-based Instance Model Generator is currently under development. When we focus on implementing or applying a model-based solution in a domain-specific context, like CPSs, we have to deal with a lack of realistic data in the first place. The key input data required for the design and development of such solutions are the models. To this end, the GAN-Based instance model generator aims to mitigate this shortage by producing a proper data set. As a GAN framework is used at the core of the generator engine, the output data (i.e. the generated Ecore-based instance models) will be structured in a realistic way (i.e. as if they were manually produced by engineers). These data might be utilised not only for testing the newly developed MDE tools but also for feeding other DL-based solutions.

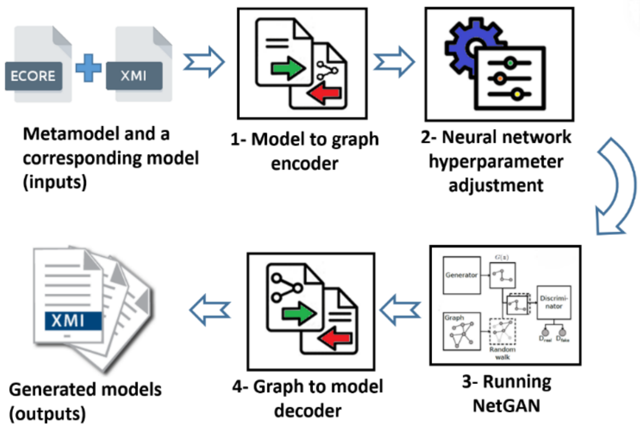

Figure 3 – Overview of the GAN-Based instance model generator framework.

Figure 3 shows an overview of the GAN-Based framework that can generate new realistic instance models in four steps. In the first step, the model-to-graph encoder extracts the structural feature of the input metamodel and based on this information creates the graph (as an adjacency matrix) related to the instance model. Next, the user should adjust the neural network hyperparameter according to the input graph features. Then, it is possible to run the training phase and after that ask the trained neural network to generate new graphs. Finally, the graph-to-model decoder converts the generated graphs to output models.

2.14.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
Instance_Generator:	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** To be decided (e.g. Eclipse Public Licence - V2.0)	This tool is still under development. The current plan is to revise the data encoding and generalise its usage by adding a configuration setting.

Table 16 Future Capabilities Implementation Roadmap of the GAN-Based Instance Model Generator Solution

2.15 Solution - DataAggregator (ROTECH)

DataAggregator is a solution composed of one or more Relational and non-Relational Databases in order to store and aggregate data for Big Data management and Machine Learning purposes (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Data Handling Capabilities (Data and Modelling)
- Storage Capabilities (Infrastructure)

2.15.1 News and Updates

The DataAggregator capabilities regarding the data handling and storage have been fully implemented. The database choice was related to the needs of TEKNE Use Case which produces homogeneous and ordered data that can be easily stored in a Relational Database.

The replication capability has also been developed using the master-slave technique[1]. In addition, we have developed a server, using Node.js, that automates the process of promoting the slave Database to master whenever the master goes offline. With this configuration we can provide business continuity to the system that relies on this Database.

The capabilities related to the elaboration of data to automate unit testing will be reported in a future WP4 deliverable.

2.15.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
Data aggregation: This solution is composed of one or more Relational and Non-relational Databases in order to store and aggregate data for Big Data management and Machine Learning purposes. Moreover, it will also include replication techniques based on the master - slave replication or any other type related to the chosen Databases.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS6 (M28) **Licence:** Licence-free	The configuration is composed of two Relational Databases connected using a replication stream. An external server does the promotion when detects a problem with the master Database. As mentioned before, further developments will be realised in the context of WP4.

Table 17 Capabilities Implementation Status of the DataAggregator Solution

[1] https://en.wikipedia.org/wiki/Replication_(computing)#DATABASE

2.16 Solution - AALpy (TUG)

AALpy (Active Automata Learning library) is a library implementing different active automata learning algorithms that support the learning of finite state models of black-box systems (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Explainability (Generic Services)
- Model-Based Capabilities (Data and Modelling)

2.16.1 News and Updates

Since the start of the project, there have been several new releases of AALpy. During the past period, they notably added new learning algorithms for both passive and active automata learning, fixed several bugs in the solution and made available different improvements to performance and usability.

2.16.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
Automata learning of deterministic systems: Inference of a behavioural model of a black-box system that behaves deterministically. Following modelling formalisms are supported: deterministic finite automata, Mealy machines, Moore machines	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** MIT	The current focus of AALpy in the context of AIDOaRt is on passive automata learning. In this domain, we are currently working on a modular framework that unifies passive learning based on state merging for DFAs, Moore machines and Mealy machines in deterministic, non-deterministic and stochastic flavours. In this respect, further developments will be realised in the context of WP4.
Automata learning of non-deterministic systems: Inference of a behavioural model of a black-box system that behaves non-deterministically. To provide efficiency, learning can also be combined with an abstraction mechanism. Following modelling formalisms are supported: Observable non-deterministic finite state machines (+ abstracted version)	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** MIT	See above.
Automata learning of stochastic systems:	**Implementation Level:** Partially Implemented	In addition to the plans regarding the unified framework, we are also working on methods for passive learning of stochastic systems that

Inference of a behavioural model of a black-box system that behaves stochastically	**Estimated Delivery Date:** MS7 (M32) **Licence:** MIT	make use of additional domain knowledge to learn hierarchic automata. Those results can probably be extended to the other automaton types as well. Once again, further developments will be realised in the context of WP4.

Table 18 Future Capabilities Implementation Roadmap of the AALpy Solution

2.17 Solution - S3D (UCAN)

S3D is a model-based System Modeling and Design framework automating several ESL design activities, e.g., the production of functional models, of performance models, or design space exploration (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)

2.17.1 News and Updates

This tool is a high-level wrapper for a number of other tools, including also the SoSIM solution (cf. Section 2.18). During the past period, the large majority of the efforts have been dedicated to make the improvements to SoSIM valuable for the TEKNE use case. Thus, up to now, the S3D launchers for SoSIM have not been developed yet. We currently plan to switch efforts into this tool during the last year of the project.

2.17.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
S3D: CPSoS modelling, analysis and design framework able to select from a library of reusable components those to be used in the current project. Using them, the functional architecture is decided. A model of the HW/SW execution platform is also developed. By mapping a functional architecture to such an execution model, different platform-specific solutions can be defined.	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS6 (M28) **Licence:** Open Source for research and academic purposes, under Licence Agreement for commercial use.	The enhancements on this tool are less relevant for the integration of UCAN solutions into the use cases. Therefore we have rescheduled the labour on this tool for the last year of the project (e.g., in the context of WP5).

Table 19 Capabilities Implementation Status of the S3D Solution

2.18 Solution - SoSIM (UCAN)

SoSIM is a solution for model-based SoS simulation allowing the simulation and performance analysis of distributed systems of systems to take architectural design and deployment decisions based on accurate performance estimations. (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)

2.18.1 News and Updates

UCAN has now defined the dataset generation methodologies for both the static technique, based on annotating basic-blocks, and the complementary dynamic technique, based on performance counters.

Regarding the static approach, the proposed technique measures the number of cycles required for a block with 1, 2, 3 and 4 instructions. Due to the lack of precision in the measurement and the fact that when executing the same group of instructions, the processor should use the same number of cycles, in order to obtain accurate data, the test group was executed multiple times, generating a sample of n repetitions of the test to be measured. In each iteration, the test was repeated four times consecutively. The objective was to measure the number of cycles required for each test. It was decided to increase the number of tests generated and perform a linear regression of the tests and measurements obtained. The linear regression returned the coefficients determining a function of the form ax + b = y, where 'a' is the value of interest since it represents the number of cycles required for the execution of the test when multiplied by four, due to the difference in speeds between the timer and the processor. This adjustment is necessary to understand how many cycles the processor requires for a particular test. To ensure the highest possible accuracy, the decision was made to double the number of tests performed again, carrying out a total of 24 tests. This was done to obtain more robust data and reduce any variability in measurements.

Regarding the dynamic approach, the code is instrumented to read the performance counters on the Intel host and measure the execution time on the target platform when executing the same code. Instrumenting the whole code of each complete benchmark will not create enough data-points for training. Furthermore, in the training phase, we want to expose a dedicated neural network to different problem sizes for each function. Data augmentation is performed on the original benchmark suite. Different combinations of the unmodified functions and argument sizes are executed while logging event counts and measuring processor time. For each combination of x,y,z functions we generate a new line on the dataset. When executing on the host it will log event counts for each combination of functions while on the host, CPU time will be recorded for that exact same function combination.

2.18.2 Capabilities Implementation Status

Capability Name & Description	Implementation Status	Comments and Release notes
SoSIM: CPSoS simulation and performance analysis tool able to simulate any of the platform-specific models defined in S3D.	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS6 (M28)	We have detected the strong dependability of accuracy with respect to the granularity of the size of the code under analysis (i.e. the phase). The NN trained in a first

Non-functional metrics such as delays, throughputs and energy consumptions can be analysed. SoSIM is able to generate execution traces that can be analysed by specific tools	**Licence:** Open Source for research and academic purposes, under Licence Agreement for commercial use.	experiment with very good results on large phases, happens not to be accurate enough in programs divided in small phases. We are still investigating how to overcome this while simplifying the methodology by linking it to a CPU simulator. We plan to continue this effort during the last year of the project in the context of WP4.

Table 20 Capabilities Implementation Status of the SoSIM Solution

2.19 Solution - HEPSYCODE (UNIVAQ)

The HEPSYCODE (HW/SW CO-DEsign of HEterogeneous Parallel dedicated Systems) solution supports hardware and software design activities such as handling non-functional requirements, specific hardware technologies, scheduling strategies, and communication between processes (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)
- Data Handling Capabilities (Data and Modelling)

2.19.1 News and Updates

Since the start of the project, HEPSYCODE has released several updates, introducing additional AI/ML algorithms for performance simulation and design space exploration. During the past period, these updates have notably addressed various bugs, enhanced overall performance, and improved the scalability of the solution. Moreover, HEPSYCODE has been expanded to support other models of computation and seamlessly integrated with tools developed by other project partners. Currently, the platform stands out for its ability to provide precise and accurate simulations through the application of advanced AI/ML techniques. It also enables rapid analysis of various architectural implementations and solutions. The integration with external tools not only speeds up modelling activities but also allows for continuous monitoring and the implementation of potential model improvements.

2.19.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
Performance Simulation and Predictions: HEPSYCODE Design Flow techniques (from high-level representation to the low level implementation), considering	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** GPL v3	HEPSYCODE allows the utilization of internal (temporary) XML files. These files are employed and used by the HEPSIM2 performance simulator. Their purpose is to store the results of the component's various functionalities by saving them in these XML files. Furthermore, additional

AI/ML techniques for simulations and predictions		AI/ML techniques have been used to enhance accuracy and reduce simulation time through the utilization of large data model collections. Future plans for MS8 (M36), to be reported in the context of WP4, involve the potential to fine-tune the AI/ML algorithms and to enhance estimations while maintaining lower simulation time.
Design Space Alternatives Exploration: HEPSYCODE Design Flow techniques (from high-level representation to the low level implementation), considering AI/ML techniques for design space exploration	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** GPL v3	The TemplateLibrary consists of XML files containing available hardware components and connections. Users can add more components by following the correct XML format. The SystemModel includes the definition of used processes and communication channels, along with their corresponding SystemC code. These files help find partitioning solutions using AI/ML techniques, aiming to improve system implementations. The output includes XML files representing different system implementations, along with simulation metrics. An implementation represents a hardware design with components and connections, including how processes are mapped to the hardware. Future plans for MS8 (M36), to be reported in the context of WP4, involve comparing different algorithms to find the best prediction models for specific case studies.
Model-Driven DevOps approach for HW/SW Co-Design: HEPSYCODE Design Flow techniques (from high-level representation to the low level implementation), considering DevOps and Standard MDE principle into an automatic co-design framework	**Implementation Level:** Partially Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** GPL v3	The language used in Hepsycode to model system behaviour, known as HML (HEPSY Modelling Language), is based on the Communicating Sequential Processes (CSP) Model of Computation (MoC). It enables the modelling of embedded applications and offers different models of computation through the ALT construct and introduced rendezvous buffers within the modelling environment. HEPSYCODE seamlessly integrates with external tracing tools, enabling continuous model refinement by monitoring modelling activities and providing suggestions for model refactoring. Additionally, it automatically translates the model into executable SystemC code through IF-MODEL-CODE-GENERATION interface, which employs Model-to-Text transformation from high-

	level Model to low-level SystemC code. Future plans for MS8 (M36), to be reported in the context of WP4, involve incorporating recommendations and the ability to analyse tracing activities for refining models. These modelling refactoring techniques aim to assist designers in their co-design activities and result in more comprehensible and self-contained application models.

Table 21 Future Capabilities Implementation Roadmap of the HEPSYCODE Solution

2.20 Solution - AsyncAPI Toolkit (UOC)

AsyncAPI Toolkit is an Eclipse-based toolset to develop Message-Driven APIs following a model-based approach starting from AsyncAPI specifications (cf. the deliverable [AIDOART-D3.2] for more details). This solution implements the following Core Tool Set components (cf. Section 3 for more details):

- Model-Based Capabilities (Data and Modelling)
- Data Handling Capabilities (Data and Modelling)

2.20.1 News and Updates

The definition and modelling of different kinds of QoS conditions for Asynchronous services has been successfully implemented during the last period. This capability extends the current OpenAPI specification with conditions over QoS metrics. The final goal is to automatically generate the appropriate monitoring rules to check these conditions / metrics (cf. Section 2.20).

2.20.2 Future Capabilities Implementation Roadmap

Capability Name & Description	Implementation Status	Roadmap and Planning
Monitoring of QoS for Asynchronous services: Automatically generates the monitoring infrastructure in order to monitor and assess the QoS for asynchronous services.	**Implementation Level:** Not Implemented **Estimated Delivery Date:** MS7 (M32) **Licence:** Eclipse Public Licence 2.0	At this point in time, this feature is still planned to be implemented by MS7 (M32). This work will be continued and reported in the context of WP4 and/or WP5.

Table 22 Future Capabilities Implementation Roadmap of the AsyncAPI Toolkit Solution

3 Mapping Solutions to AIDOaRt Core Tool Set Components

In this section, we present the final and up-to-date list of solution components realising the functional specification of the Core Tool Set components. During the whole duration of WP3, this mapping has been defined and then iteratively refined by the Solutions Providers based on the potential of their proposed solutions in fulfilling the description, specification, and functional interfaces of each component of the AIDOaRt Framework Architecture.

3.1 Mapping to Storage Capabilities

In this section, we present the list of solution components realising the "Storage Capabilities" component of the Core Tool Set. Each solution name is followed by the partner acronym between parentheses. The full partners names can be found in the Partners Acronyms table in the preamble of this document.

Solution Name	Rationale
Infrastructure as Code (IaC) expertise (AND)	AND can help set up data storage solutions through IaC means. We have expertise in setting up both larger and smaller data storage solutions.
devmate (AST)	Devmate contributes to this component through its core functionalities of handling data in non-volatile storage.
DataAggregator (ROTECH)	Storage capabilities can be achieved by using different databases configured according to the end user’s needs.
Constellation (SOFT)	Constellation allows to store and retrieve, in SVN repositories, software artefacts related to Modelio models.

Table 23 Solutions Mapping to the "Storage Capabilities" Component

Solution Name	Rationale
Infrastructure as Code (IaC) Expertise (AND)	AND can help set up data storage solutions through IaC means. We have expertise in setting up both larger and smaller data storage solutions.
devmate (AST)	devmate can store its test model and usage data on disk using json text and an SQL database
DataAggregator (ROTECH)	DataAggregator provides a set of tools that physically store various kind of data in a Database.
Constellation (SOFT)	Constellation allows to store, in SVN repositories, software artefacts related to Modelio models.

Table 24 Solutions Mapping to the "IF-INFRA-STORING-ARTEFACTS" Interface

Solution Name	Rationale
Infrastructure as Code (IaC) Expertise (AND)	AND can help set up data storage solutions through IaC means. We have expertise in setting up both larger and smaller data storage solutions.

devmate (AST)	devmate can load its test model and usage data from disk using json text and an SQL database
DataAggregator (ROTECH)	DataAggregator provides a set of tools retrieve various kind of data stored in a Database.
Constellation (SOFT)	Constellation allows to retrieve, from hosted SVN repositories, software artefacts related to Modelio models.

Table 25 Solutions Mapping to the "IF-INFRA-RETRIEVING-ARTEFACTS" Interface

There are different available solutions targeting the support for storing the data/models to be created and/or handled when applying the AIDOaRt overall solution in practice. They are ranging from Cloud-based solutions to more traditional database ones, including dedicated model repositories. Moreover, there are several industrial partners having a solid experience in setting up such storage infrastructure(s). This expertise is very valuable in the context of the AIDOaRt Core Tool Set.

3.2 Mapping to Computation Capabilities

In this section, we present the list of solution components realising the "Computation Capabilities" component of the Core Tool Set. Each solution name is followed by the partner acronym between parentheses. The full partners names can be found in the Partners Acronyms table in the preamble of this document.

Solution Name	Rationale
Cloud expertise (AND)	AND can help with provisioning computational power in the cloud. Be it in Kubernetes or through other options provided by cloud providers.
Kolga (AND)	Kólga allows for running tasks such as building, testing and deployments. It can automatically provision new instances of an application and run jobs based on new deployments.
a2k-modev (ITI)	Computations for analysis and simulation are performed on servers managed by the A2K framework.
MOMOT (JKU)	MOMOT offers computation capabilities in terms of optimization of in-place model transformations via SEO (current version) and reinforcement learning algorithms (an AIDOaRt extension).

Table 26 Solutions Mapping to the "Computation Capabilities" Component

Solution Name	Rationale
Kolga (AND)	Kólga allows for running tasks such as building, testing and deployments. It can automatically provision new instances of an application and run jobs based on new deployments.
Cloud Expertise (AND)	AND can help with provisioning computational power in the cloud. Be it in Kubernetes or through other options provided by cloud providers.
a2k-modev (ITI)	The a2k/analyser provides several performance analysis services. In particular we use compositional performance timing analysis to determine the expected

	maximum response times for tasks with precedence constraints. The a2k/analyser service also provides a sensitivity analysis component for determination of CPU load factors.
MOMOT (JKU)	MOMoT gets a given instance model (that represents an optimization problem), its respective metamodel, and related Henshin rules, then can find the proper orchestration of transformation rules to optimise.

Table 27 Solutions Mapping to the "IF-INFRA-COMPUTING-FROM-ARTEFACTS" Interface

There are different available solutions targeting the support for performing different kinds of computations over the data/models to be created and/or handled when applying the AIDOaRt overall solution in practice. Such computation capabilities are also needed when running the AIDOaRt continuous development process over a concrete CPS. Interestingly, several partners (notably industrial ones) are highly experienced in providing and working with cloud-based computing infrastructures and related services. We rely on this experience to provide the required computation capabilities in the context of the AIDOaRt Core Tool Set.

3.3 Mapping to Data Handling Capabilities

In this section, we present the list of solution components realising the "Data Handling Capabilities" component of the Core Tool Set. Each solution name is followed by the partner acronym between parentheses. The full partners names can be found in the Partners Acronyms table in the preamble of this document.

Solution Name	Rationale
Cloud expertise (AND)	AND can help with setting up data pipelines for storing, managing and analysing data in the cloud. We have experience from multiple cloud providers for how to ingest and handle larger amounts of data.
devmate (AST)	Devmate contributes to this component through its core functionalities of handling internal data.
ATL (IMTA)	If the considered data is expressed as model(s), ATL provides the core data handling capability of specifying and executing data transformations. Such data transformations can be defined based on any set of source and target data sets (as long as they can be expressed as models).
EMF Views (IMTA)	If the considered data is expressed as model(s), EMF Views provides the core data handling capability of specifying and building views over (possibly large) sets of heterogeneous data sets. Such views conform to viewpoints expressed on the corresponding data set formats (that can be expressed as metamodels).
a2k-runman (ITI)	New methods for handling the incoming monitoring data in real time will be developed.
DataAggregator (ROTECH)	DataAggregator will be capable of handling all data format compatible with OSI Layer 7 in order to save, retrieve and manipulate them.
HEPSYCODE (UNIVAQ)	UNIVAQ will realises data handling capabilities with the integration of a well-know HW/SW Co-Design methodology able to capture system behaviours from input data set (from WP2) and produce deployment solutions applying

	AI/ML techniques in WP4 by means of MDE models, system-level simulation and AI algorithms.
AsyncAPI Toolkit (UOC)	AsyncAPI toolkit will provide the capabilities to save and retrieve the monitored QoS of Asynchronous services.
TemporalEMF (UOC)	TemporalEMF provides the capability to load, navigate, query and save instances of models (i.e. data) with temporal capabilities.

Table 28 Solutions Mapping to the "Data Handling Capabilities" Component

Solution Name	Rationale
Cloud Expertise (AND)	AND can help with setting up data pipelines for storing, managing and analysing data in the cloud. We have experience from multiple cloud providers for how to ingest and handle larger amounts of data.
devmate (AST)	devmate has capabilities to load and deserialise (JSON) data from interfaces, memory or disk.
EMF Views (IMTA)	EMF Views allows loading different data models that contribute to the different model views it allows to specify and build.
ATL (IMTA)	ATL allows loading the (input) data models and (input/output) data metamodels that are possibly required by the different model-to-model transformations it allows to specify and build.
a2k-runman (ITI)	The component collects historical run-time data and uses this for anomaly detection.
DataAggregator (ROTECH)	DataAggregator provides a set of tools to load in memory different kinds of data for further manipulation.
HEPSYCODE (UNIVAQ)	The goal is to load the TemplateLibrary and the SystemModel. The TemplateLibrary is a set of xml files containing the available hardware basic blocks and physical channels; it can be expanded by the user by following the correct xml format. The SystemModel contains the definition of all used CSP (Communicating Sequential Processes) and CSP channels together with the corresponding SystemC code.
TemporalEMF (UOC)	TemporalEMF loads temporal information from data models that is stored in a persistence backend thanks to its own persistence manager.

Table 29 Solutions Mapping to the "IF-DATA-LOADING" Interface

Solution Name	Rationale
devmate (AST)	devmate's test model (and other data) can be displayed and modified through the user interface or a JSON structure
EMF Views (IMTA)	Model views built with EMF Views are virtual models possibly interrelating several data models which have to be navigated.
DataAggregator (ROTECH)	DataAggregator provides a set of tools to easily navigate the data after loading.
TemporalEMF (UOC)	TemporalEMF leverages the EMF framework to enable users to navigate through the elements of temporal information regarding a data model.
AsyncAPI Toolkit (UOC)	AsyncAPI enables users to navigate between single connected instances of messaged data.

Table 30 Solutions Mapping to the "IF-DATA-NAVIGATION" Interface

Solution Name	Rationale

devmate (AST)	devmate's test model (and other data) can be displayed and modified through the user interface or a JSON structure
ATL (IMTA)	ATL allows to specify and run, as model transformations, different queries on one or several input data models and data metamodels.
DataAggregator (ROTECH)	DataAggregator provides a set of tools to query the data.
TemporalEMF (UOC)	TemporalEMF provides the capability to perform temporal queries on historical and current data models' information.
AsyncAPI Toolkit (UOC)	AsyncAPI provides the capability to perform queries on single instances on messaged data.

Table 31 Solutions Mapping to the "IF-DATA-QUERYING" Interface

Solution Name	Rationale
Cloud Expertise (AND)	AND can help with setting up data pipelines for storing, managing and analysing data in the cloud. We have experience from multiple cloud providers for how to ingest and handle larger amounts of data.
devmate (AST)	devmate has capabilities to save and serialise (JSON) data of its model from interfaces, memory, disk and save data to a SQL database.
EMF Views (IMTA)	EMF Views allows to save back the data models (with some restrictions) that contribute to the different model views it allows to specify and build.
ATL (IMTA)	ATL allows to save the (output) data models that are possibly produced by the different model-to-model transformations it allows to specify and build.
DataAggregator (ROTECH)	DataAggregator provides a set of tools to save data in memory in different kinds of formats.
HEPSYCODE (UNIVAQ)	The goal is to store the results of the various functionalities of the component by saving them in XML files It generates a set of xml files representing different resulting implementations for the system, along with a characterization of each implementation in terms of simulation metrics An implementation is intended as a HW architecture (basic blocks and interconnects) along with the mapping of the processes to such hardware.
TemporalEMF (UOC)	TemporalEMF saves temporal information of data models to a persistence backend.

Table 32 Solutions Mapping to the "IF-DATA-SAVING" Interface

There is a significant number of available solutions targeting the support for handling the data/models to be considered when applying the AIDOaRt overall solution in practice. Some of these are traditional data handling solutions (potentially covering a wide range of data formats), some others are model-based solutions that can be particularly relevant in a data handling context (as data can be very often expressed as structured models). We rely on all these solutions and on the expertise of the corresponding project partners to provide the required scalable data handling capabilities in the context of the AIDOaRt Core Tool Set (cf. also the deliverable [AIDOART-D2.3]).

3.4 Mapping to Model-Based Capabilities

In this section, we present the list of solution components realising the "Model-Based Capabilities" component of the Core Tool Set. Each solution name is followed by the partner acronym between parentheses. The full partners names can be found in the Partners Acronyms table in the preamble of this document.

Solution Name	Rationale
ESDE (ACO)	ESDE enables the specification of software, and especially embedded one, as a SystemC executable model. Moreover, this model serves as a specification to automatically generate embedded software that can be targeted to a virtual platform (VP) model to build up a platform dependent model (PDM). In this sense, ESDE supports model transformation. Models are stored as code, i.e., SystemC+C/C++ for the high-level eSW model; and targeted to C++ embedded code, which together with the virtual platform description (scripts, C, etc) lead to an executable PDM. In this sense, ESDE does handle models as queryable and navigable databases.
devmate (AST)	Devmate contributes to this component through its core functionalities of model handling and test code generation.
ATL (IMTA)	ATL provides the core model-based capability of specifying and executing model-to-model transformations. Such model transformations can be defined based on any set of source and target metamodels / languages.
EMF Views (IMTA)	EMF Views provides the core model-based capability of specifying and building model views over (possibly large) sets of heterogeneous models. Such views conform to viewpoints expressed on the corresponding metamodels, and can be handled as any regular model.
a2k-depman (ITI)	Modelling of system, application and execution architectures. Multi-objective optimisation for exploration of design and execution spaces.
a2k-modev (ITI)	A2K enables modelling of a heterogeneous, distributed system platform, application software, and its execution environment.
AutomationML Modelling (JKU)	AutomationML Modelling is a suite of model-driven tools (metamodels, editors, UML profiles) based on the AutomationML standard (www.automationml.org).
DevOpsML (JKU)	DevOpsML allows the modelling of the generic architecture of a DevOps platform in terms of Tools and Interfaces, provided/required Capabilities, and addressed Concerns. Tools, Interfaces, Capabilities, and Concerns are chosen in order to (partially) support an arbitrary DevOps Process, which can be described by a given Software Process Description Language (SPDL) like SPEM or BPMN or any user-defined DSL. The approach is based on RMF-based technologies.
GAN-Based Instance Model Generator (JKU)	This solutions provides support for automatically generating realistic models from a given metamodel. It is based on a trained GAN that is used to predict realistic model elements and their relations in the generated models.
JSON Schema DSL (or MDE4JSON) (JKU)	JSON Schema DSL (a.k.a. MDE4JSON) allows the management of JSON artefacts as models via Eclipse EMF-based technologies.
MOMOT (JKU)	MOMOT provides its own internal DSL to specify the optimization problem.
Modelling Process Mining Tool (JKU)	Modelling Process Mining Tool provides Model-based Process Mining capabilities to Sirius-based graphical editors. The tool relies on the XES IEEE Standard (https://xes-standard.org/) and provides an EMF-based

	metamodels of the original XES XML Schema, making it compatible with EMF-based technologies.
Modelio (SOFT)	Almost all of the existing Modelio services and capabilities (e.g., modelling environment and editors, model repository, document generators) would contribute to this AIDOaRt component.
AALpy (TUG)	The AALpy library supports different modelling formalisms including loading, storing and visualisation of deterministic, non-deterministic and stochastic automata as well as running tests on them.
S3D (UCAN)	The tools suite in S3D (MAST) will allow performance estimations even at the analysis phase by using temporal budgets of initial behavioural models. Response times will be linked to behavioural models for concrete analysis contexts, this enables design space exploration and further optimization of design parameters.
SoSIM (UCAN)	This tool will handle and generate performance indicators that can be improved in design space exploration optimization processes.
HEPSYCODE (UNIVAQ)	UNIVAQ will realise model-based capabilities by means of MDE approaches and principles applied to HW/SW Co-Design methodologies.
AsyncAPI Toolkit (UOC)	AsyncAPI toolkit provides the capability to load, navigate, save and transform AsyncAPI based models (EMF, YAML, JSON). Furthermore, AsyncAPI toolkit can be used to generate boilerplate java code from the defined model.
TemporalEMF (UOC)	TemporalEMF provides the capability to load, navigate, query and save models with temporal capabilities.
WAPIml (UOC)	WAPIml is an Eclipse-based editor for OpenAPI. It provides the capability to load, navigate, save and transform OpenAPI-based models (EMF, YAML, JSON).

Table 33 Solutions Mapping to the "Model-Based Capabilities" Component

Solution Name	Rationale
ESDE (ACO)	ESDE does not handle application or virtual platform models as a database (e.g., under EMF), that needs to be loaded in some specific framework for later model-based activities. However, ESDE handles models (whole system application model, and virtual platform model) as code fed as inputs and, in that sense, “loaded” for the different enabled design activities. In ESDE, the SystemC application model is taken as an input for the different design activities, i.e., executable model generation and for software generation. Similarly, the virtual platform (VP) model is compiled and dynamically loaded for building the overall platform dependent model before running the VP-based simulation.
devmate (AST)	devmate can load its (JSON serialised) test model from memory, through an interface or disk.
EMF Views (IMTA)	EMF Views allows loading the models and metamodels that contribute to the different model views it allows to specify and build.
ATL (IMTA)	ATL allows loading the (input) models and (input/output) metamodels that are required by the different model-to-model transformations it allows to specify and build.

a2k-modev (ITI)	The a2k/modelling service enables loading of system models from long-term storage, including version control via GIT.
a2k-depman (ITI)	The a2k-depman component uses the a2k/modelling service to enable loading of system models from long-term storage, including version control via GIT.
DevOpsML (JKU)	DevOpsML is a model-driven approach to describe DevOps processes, DevOps platforms, and their combination via EMF-base process, platform, and linking models. Model-Loading is based on EMF-based API generated by the provided metamodels.
JSON Schema DSL (or MDE4JSON) (JKU)	JSONSchemaDSL is a model-driven approach to manage JSON artefacts (metaschema, schemas, and schema instances) as EMF-based models. Xtext-based textual editors and OCL constraints are automatically generated. Model-Loading is based on EMF-based API generated by the provided and generated metamodels.
AutomationML Modelling (JKU)	AutomationML Modelling is a suite of model-driven tools (metamodels, editors, UML profiles) based on the AutomationML standard (www.automationml.org). Model-Loading is based on an EMF-based API provided by AutomationML Modelling.
Modelio (SOFT)	Modelio natively supports several import formats such as XMI, Reqlf, BPMN, etc. It also provides a Java API which allows the development of custom imports such as XML, JSON, etc.
AALpy (TUG)	AALpy supports (de-)serialising behavioural models for different types of automata.
TemporalEMF (UOC)	TemporalEMF is an EMF-based model manager that loads data models that are stored in a persistence backend thanks to its own persistence manager.
AsyncAPI Toolkit (UOC)	AsyncAPI is able to load a stored model of asynchronous APIs.
WAPIml (UOC)	WAPIml is able to load a stored model representing OpenAPI definitions of RESTful services.

Table 34 Solutions Mapping to the "IF-MODEL-LOADING" Interface

Solution Name	Rationale
ESDE (ACO)	In general, any SystemC model, and specifically the one generated in ESDE, supports its own introspection. This is the same for Virtual Platform models in the OVP environment. Being C/C++-based, a programmatic access to specific model elements on a set of them is possible. This introspection capability is being used to set up monitors for trace generation.
EMF Views (IMTA)	Model views built with EMF Views are virtual models interrelating several models and/or metamodels which have to be navigated.
a2k-modev (ITI)	The a2k/modelling service enables loading of system models from long-term storage, including version control via GIT.
DevOpsML (JKU)	DevOpsML is a model-driven approach to describe DevOps processes, DevOps platforms, and their combination via EMF-base process, platform,

	and linking models. Model-Navigation is based on EMF-based API generated by the provided metamodels.
JSON Schema DSL (or MDE4JSON) (JKU)	JSONSchemaDSL is a model-driven approach to manage JSON artefacts (metaschema, schemas, and schema instances) as EMF-based models. Xtext-based textual editors and OCL constraints are automatically generated. Model-Navigation is based on EMF-based API generated by the provided and generated metamodels.
AutomationML Modelling (JKU)	AutomationML Modelling is a suite of model-driven tools (metamodels, editors, UML profiles) based on the AutomationML standard (www.automationml.org). Model navigation is based on an EMF-based API provided by AutomationML Modelling.
Modelio (SOFT)	Once a model is loaded by Modelio, it is possible to use both its graphical interface or its Java API to navigate/browse through the model.
TemporalEMF (UOC)	TemporalEMF leverages the EMF framework to enable users to navigate through the elements of a data model.
AsyncAPI Toolkit (UOC)	AsyncAPI enables users to navigate through the elements of a model of asynchronous APIs.
WAPIml (UOC)	WAPIml facilitates the navigation of OpenAPI definitions in a model-based environment.

Table 35 Solutions Mapping to the "IF-MODEL-NAVIGATION" Interface

Solution Name	Rationale
devmate (AST)	devmate's internal models can be queried through internal API methods and the user interface.
ATL (IMTA)	ATL allows to specify and run, as model transformations, different queries on one or several input models and metamodels.
a2k-modev (ITI)	The a2k/modelling service enables the user to interrogate the model and search for specific items.
Modelio (SOFT)	All model elements under Modelio are accessible through its Java API. Modelio provides several ways for querying a model from a Python script to the development of a dedicated extension.
AALpy (TUG)	Behavioural models in AALpy are instances of classes corresponding to the specific automaton type and can be queried and transformed.
TemporalEMF (UOC)	TemporalEMF provides the capability to perform temporal queries on historical and current data models' information.
AsyncAPI Toolkit (UOC)	AsyncAPI toolkit enables users to perform queries on model elements based on the tool’s metamodel.
WAPIml (UOC)	WAPIml enables users to perform queries on the models of OpenAPI definitions of RESTful services.

Table 36 Solutions Mapping to the "IF-MODEL-QUERYING" Interface

Solution Name	Rationale
devmate (AST)	devmate transforms its test model to various other models. This includes models used by other micro-services and older versions of the test model.

ATL (IMTA)	ATL allows to specify and run different model-to-model transformations on one or several input models and metamodels to produce one or several output models.
a2k-modev (ITI)	The a2k/modelling service provides several graphical editors to create, transform, and manipulate the system model. Model transformation is also provided in conjunction with the a2k/analyser and a2k/optimiser services which perform design space exploration and optimisation.
a2k-depman (ITI)	The a2k/optimiser service in conjunction with the a2k/analyser service in the a2k-modev component provides facilities for design space exploration and optimisation of system architectures.
MOMOT (JKU)	MOMOT can be used for in-place model transformation to find the orchestration of executing transformation roles to reach the optimised form of the input(initial) model based on defined objectives.
AutomationML Modelling (JKU)	The AutomationML modelling suite provides generic model transformations from/to SysML models. Additional rules are needed based on modelling guidelines of interface users.
Modelio (SOFT)	Model transformation, under Modelio, can only be done through a dedicated Java or Python development. This development will use Modelio API to define how a loaded Modelio model can be transformed into another model.
AALpy (TUG)	Behavioural models in AALpy are instances of classes corresponding to the specific automaton type and can be queried and transformed.
S3D (UCAN)	Among the internal utilitarian tools that S3D holds and maintains, MSSyn (Model for Simulation Synthesis) is the one dedicated to transform the general MARTE/UML model of an application, properly annotated, into a versatile simulation model meant for performance and power consumption simulation. This tool plays a model transformation role in the context of the AIDOART Framework Architecture.
AsyncAPI Toolkit (UOC)	AsyncAPI could perform predefined transformations from annotated EMF and UML models to its own API metamodel.
WAPIml (UOC)	WAPIml permits updating OpenAPI definitions in any UML2-compatible modeller.

Table 37 Solutions Mapping to the "IF-MODEL-TRANSFORMATION" Interface

Solution Name	Rationale
EMF Views (IMTA)	Model views built with EMF Views allows to interrelate/federate several different models and metamodels for different purposes (e.g. traceability, dependence).
Modelio (SOFT)	By default, Modelio provides a parameterisable view named "link editor". This later allows the layout of any kind of existing relations between elements from possibly any kind of formalism. Modelio also provides the ability to aggregate specific parts of models (called fragments) into different projects in order to federate these models.

Table 38 Solutions Mapping to the "IF-MODEL-TRACEABILITY-FEDERATION" Interface

Solution Name	Rationale

devmate (AST)	devmate has capabilities to save its internal models to memory or disk, serialised in JSON.
EMF Views (IMTA)	EMF Views allows saving back the models and metamodels (with some restrictions) that contribute to the different model views it allows to specify and build.
ATL (IMTA)	ATL allows to save the (output) models that are produced by the different model-to-model transformations it allows to specify and build.
a2k-modev (ITI)	The a2k/modelling service component enables saving of models to long-term storage, including version control via GIT.
a2k-depman (ITI)	The component enables saving of models to long-term storage, including version control via GIT.
DevOpsML (JKU)	DevOpsML is a model-driven approach to describe DevOps processes, DevOps platforms, and their combination via EMF-base process, platform, and linking models. Model-Saving is based on EMF-based API generated by the provided metamodels.
JSON Schema DSL (or MDE4JSON) (JKU)	JSONSchemaDSL is a model-driven approach to manage JSON artefacts (metaschema, schemas, and schema instances) as EMF-based models. Xtext-based textual editors and OCL constraints are automatically generated. Model-Saving is based on EMF-based API generated by the provided and generated metamodels.
AutomationML Modelling (JKU)	AutomationML Modelling is a suite of model-driven tools (metamodels, editors, UML profiles) based on the AutomationML standard (www.automationml.org). Model-Saving is based on an EMF-based API provided by AutomationML Modelling.
Modelio (SOFT)	Modelio natively supports several import/export formats such as XMI, ReqIf, BPMN, etc. These formats can be used by default for model serialisation. Custom formats can also be implemented in Java.
AALpy (TUG)	AALpy supports (de-)serialising behavioural models for different types of automata.
HEPSYCODE (UNIVAQ)	The language introduced in Hepsycode for modelling system behaviour, called HML (HEPSY Modelling Language), is based on the Communicating Sequential Processes (CSP) Model of Computation (MoC). It allows modelling the behaviour of the system as a network of processes communicating through unidirectional synchronous channels. Using HML, it is possible to specify the System Behaviour Model (SBM), an executable model of the system behaviour, a set of Non Functional Constraints (NFC) and a set of Reference Inputs (RI) used for simulation-based activities.
TemporalEMF (UOC)	TemporalEMF is an EMF-based model manager that saves data models to a persistence backend.
AsyncAPI Toolkit (UOC)	AsyncAPI enables users to save a model of their asynchronous APIs.
WAPIml (UOC)	WAPIml serialises OpenAPI models in JSON and YML formats.

Table 39 Solutions Mapping to the "IF-MODEL-SAVING" Interface

Solution Name	Rationale
GAN-Based Instance Model Generator (JKU)	GAN-based Instance Model Generator is able to learn the structure of its input model and then generate new realistic instance models in XMI format.
ESDE (ACO)	In ESDE embedded software code generation from the SystemC model is performed. This generation consists of a re-interpretation of the input SystemC code into a targeted embedded code. The host-specific implementation for a simulation target (represented by the SystemC library) is substituted by calls to target specific RTOS services and I/O drivers. The reinterpretation aims to preserve a common executive semantics, as provided by the SystemC Language Reference Manual (LRM).
devmate (AST)	devmate can produce code (structured text) from its internal model.
a2k-modev (ITI)	The a2k/simulation service can create template source code files (e.g. for run-time monitoring). It also provides code generation for simulation.
Modelio (SOFT)	Several Modelio extensions for code modelling (Java, C#, C++) exist. All of them include dedicated code generation. Any other code generation will require a dedicated Java development.
S3D (UCAN)	S3D holds and maintains a number of internal utilitarian tools. ESSyn (Embedded Systems Synthesis) is an elaborated Model-to-text transformation that works as a code generator. This tool operates with the initial input to the S3D simulation tool and is able to clean the initial model from performance-oriented annotations in order to extract, prepare, and compile the final application for a target platform.
SoSIM (UCAN)	SoSIM allows to simulate the expected performance of an application in the consideration of its deployment over a concrete hardware platform. As part of its final deployment stage it encompasses a code generator for such specific processing unit. In the aim of taking profit from cross-platforms simulations UCAN is also exploring the transformation of the code to alternative platforms.
HEPSYCODE (UNIVAQ)	This interface offers a Model-to-Text transformation from High-Level Model to low-level Systemc code. The system behaviour is enclosed in a single SC_MODULE containing all the CSP processes (SC_THREAD) and the CSP channels (sc_csp_channel). Other SC_MODULE and sc_csp_channel objects are then used to model the Test-Bench (i.e., STIMULUS and DISPLAY) and are connected to the system through appropriate SC_PORT objects. CSP processes are modeled using the classic SC_THREAD while CSP channels were modeled by introducing a proper sc_csp_channel. A "CSP SC_THREAD" has an init section and an infinite loop behaviour, while it only accesses its local variables and therefore communicates with other "CSP SC_THREAD" only through CSP channels. Internal (temporary) xml files are used (and created) by the simulator.
AsyncAPI Toolkit (UOC)	AsyncAPI automatically generates API code from an EMF/UML model.
WAPIml (UOC)	WAPIml enables the use of 3rd party OpenAPI code generators in a model-based environment, thus enabling it to automatically generate code from OpenAPI definitions of RESTful services.

Table 40 Solutions Mapping to the "IF-MODEL-CODE-GENERATION" Interface

Solution Name	Rationale
Modelling Process Mining Tool (JKU)	Modelling Process mining tool is capable of logging and collecting user interaction during modelling activities.

Table 41 Solutions Mapping to the "Modelling Process Tracing" Interface

There is a very large number of available solutions targeting the support for creating, and handling all the different kinds of models to be considered when applying the AIDOaRt overall solution in practice. As we have seen before with the Data Handling component, such solutions can be particularly useful for more easily manipulating the data as structured models. These model-based solutions are also necessary for dealing with all the other kinds of models that are created or already existing in system development processes (e.g., domain models, software and system design models, development models, process models). Based on this large set of solutions, we can address the needs of the use case providers in terms of model-based capabilities in the context of the AIDOaRt Core Tool Set. Additionally, we can rely on the partners' long-term experience in the domain to identify and use external model-based solutions if required.

3.5 Mapping to Accountability

In this section, we present the list of solution components realising the "Accountability" component of the Core Tool Set. Each solution name is followed by the partner acronym between parentheses. The full partners names can be found in the Partners Acronyms table in the preamble of this document.

Solution Name	Rationale
ESDE (ACO)	From a generic anomaly detection capability, fused with pattern-based analysis, ACORDE aims to shed further light on the description of causality relationships among the detected relevant anomalies. A proper presentation and/or visualisation of these relations (likely representable via a DAG) should contribute to deep and understandable explanation of the failure roots, and on the usage of the found patterns for early pattern-based detection of such bugs/failures.
DTsynth (AIT)	This interface provides analysis results from DTsynth.
INT-XAI (INT)	INT-XAI determines and visualizes the marginal contribution of a feature/part of an image towards a model output.

Table 42 Solutions Mapping to the "IF-GENSERV-GETTING-ANALYSIS-RESULTS" Interface

Solution Name	Rationale
DTsynth (AIT)	DTSynth allow performing different kinds of analysis over models and CPS.
INT-XAI (INT)	INT-XAI provides the possibility to validate an AI-based model in terms of explainability. It applies to object detection related to computer vision and image processing.
AALpy (TUG)	AALpy interfaces with model-checkers and visualisation tools that can be used for manual and (semi-)automatic analysis of behavioural models.

Table 43 Solutions Mapping to the "IF-GENSERV-ANALYSING-ARTEFACTS" Interface

There is a decent coverage by different solutions targeting the support for accountability, as far as the use of AI-based solutions is concerned. Please note that the work around this component is strongly linked to the work that has been and will be performed in the context of WP4 regarding the AI-Augmented Tool Set. Thus, more details on these solutions (and related ones) can also be found in the WP4 deliverables.

3.6 Mapping to Explainability

In this section, we present the list of solution components realising the "Explainability" component of the Core Tool Set. Each solution name is followed by the partner acronym between parentheses. The full partners names can be found in the Partners Acronyms table in the preamble of this document.

Solution Name	Rationale
ESDE (ACO)	An interesting feature that ACORDE would like to explore in AIDOaRt is the possibility to obtain the patterns leading to detection of functional and performance anomalies and flaws. It is expected that those patterns can involve activations of certain event sequences at different layers with specific characteristics (e.g., certain HW interrupts, at certain rate, with certain stack configuration, leading to uncontrolled stack growth, and other consequences at application level). If, for instance, these patterns are well identified after training time, and a human readable explanation can be assigned, it is envisaged that the detection can retrieve such explanations and help designers to locate and fix issues detected.
DTsynth (AIT)	DTsynth uses passive and/or active learning and puts a focus on understandability and trustability. To achieve this, explainable AI methods are planned to be used, potentially in combination with automata learning approaches.
INT-XAI (INT)	INT-XAI is a solution that brings the concept of explainability to the deep learning modelling world, giving an additional validation method. It is designed to assist users during their modelling activities by giving a humanly interpretable reason for the model output. This component enables the monitoring of an AI model by giving a measure of its reliability. In the context of the specific use case, it provides an important hint on what aspects of the objects (vehicles, pedestrians, etc.) are important to be recognized.
AALpy (TUG)	AALpy allows to extract behavioural models of black-box systems which allows for easier understanding and enables verifying (simpler to understand) properties of those systems via model checking.

Table 44 Solutions Mapping to the "Explainability" Component

Solution Name	Rationale
ESDE (ACO)	From a generic anomaly detection capability, fused with pattern-based analysis, ACORDE aims to shed further light on the description of causality relationships among the detected relevant anomalies. A proper presentation and/or visualisation of these relations (likely representable via a DAG) should contribute to deep and understandable explanation of the failure roots, and on the usage of the found patterns for early pattern-based detection of such bugs/failures.

DTsynth (AIT)	This interface provides analysis results from DTsynth.
INT-XAI (INT)	INT-XAI determines and visualizes the marginal contribution of a feature/part of an image towards a model output.

Table 45 Solutions Mapping to the "IF-GENSERV-GETTING-ANALYSIS-RESULTS" Interface

Solution Name	Rationale
DTsynth (AIT)	DTSynth allow performing different kinds of analysis over models and CPS.
INT-XAI (INT)	INT-XAI provides the possibility to validate an AI-based model in terms of explainability. It applies to object detection related to computer vision and image processing.
AALpy (TUG)	AALpy interfaces with model-checkers and visualisation tools that can be used for manual and (semi-)automatic analysis of behavioural models.

Table 46 Solutions Mapping to the "IF-GENSERV-ANALYSING-ARTEFACTS" Interface

There is a decent coverage by different solutions targeting the support for explainability, as far as the use of AI-based solutions is concerned. Please note that the work around this component is strongly linked to the work that has been and will be performed in the context of WP4 regarding the AI-Augmented Tool Set. Thus, more details on these solutions (and related ones) can also be found in the WP4 deliverables.

4 Mapping Use Case Requirements to AIDOaRt Core Tool Set Components

In this section, we present the final and up-to-date mapping of the use case requirements and data requirements to the Core Tool Set components. During the whole duration of WP3, this mapping has been defined and then iteratively refined by the Case Study Providers based on the potential of each component of the AIDOaRt Framework Architecture in satisfying each of their use case requirements and data requirements.

For the sake of clarity, we present these mappings per component. For each component, we list the correlated requirements in a separate section for each component, grouped in two tables. The first table lists the related use case requirements, and the second table lists the related use case data requirements.

Note that each requirement identifier is prefixed by the partner acronym. The full partners names can be found in the Partners Acronyms table in the preamble of this document.

4.1 Mapping to Storage Capabilities

In this section, we present the mapping of the use case requirements and data requirements to the "Storage Capabilities" component of the Core Tool Set.

Requirement ID	Requirement Description	Rationale
VCE_R05	Customise standards based modelling frameworks (e.g. UAF, SysML, UML) and metamodels to develop system, software, data architecture models	An infrastructure for storage components is expected to ensure storage and retrieval of customised standards-based modelling frameworks and metamodels for architecture design and modelling. VCE would use storage to store the internal customised modelling frameworks for reuse and easy access.
VCE_R07	Development of standard data classification, reusable definition, representation, usage	An infrastructure for storage components is expected to ensure that the data stored is represented in an interoperable format that allows for reuse and interchange with third party tools and standards. VCE would use storage to store and retrieve artefacts or data in interoperable formats to enable third party tool interchange.
BT_R01	NLP contextual analysis of requirements and match against database of responses/solutions	The Storage Component is expected to ensure saving and retrieval of requirements used in the previous projects as well as in further development. Alstom would use the storage component to store the previously

		processed requirements used for the training AI/ML model.
BT_R02	ML aided control model parameterization during propulsion system testing	Alstom will use the Storage Capabilities Component to store the sensors' data and the models' parameters.
W_R_3	Extract data from steps into DevOps process.	Data collected is stored and managed.
W_R_4	Log file storing, indexing, searching, clustering and comparing	Data collected is stored and managed.
W_R_2	Quality monitoring and predictions in devops process	In order to do quality monitoring, some data needs to be stored.

Table 47 Use Case Requirements Mapping to the "Storage Capabilities" Component

Data Requirement ID	Data Requirement Description	Rationale
AVL_SEC_DR01	CAN data of a realistically behaving vehicular power train to train a plausibility model ANN	The storage component must provide safe and secure storage (authenticity, integrity and availability must be assured).
AVL_SEC_DR02	CAN data of a realistically behaving vehicular power train to train an anomaly detection ANN	The storage component must provide safe and secure storage (authenticity, integrity and availability must be assured).
W_DR_02	To identify non-trivial indicators for quality shortcomings, the test cases could be parsed with NLP.	The data collected is stored and managed.
W_DR_03	To identify non-trivial indicators for quality shortcomings, the source code and recent changes could be parsed.	Data collected is stored and managed.
W_DR_05	To identify non-trivial indicators for quality shortcomings, the test scripts could be parsed.	The data collected is stored and managed.
W_DR_07	To identify non-trivial indicators for quality shortcomings, the logs from static code analysis could be parsed.	The data collected is stored and managed.
W_DR_08	To identify non-trivial indicators for quality shortcomings, the compilation logs could be parsed.	The data collected is stored and managed.
W_DR_09	To identify anomalies or gradual degradation in performance, the test execution logs could be parsed	The data collected is stored and managed.
W_DR_10	To identify anomalies or gradual degradation in performance, as well as functional issues and error messages, the device communication logs should be parsed.	The data collected is stored and managed.

W_DR_11	To extract information on non-functional characteristics (from e.g. free, top, etc.), the device communication logs should be processed.	The data collected is stored and managed.
W_DR_12	To identify the current configuration of a device being tested, the device communication logs could be processed.	The data collected is stored and managed.
W_DR_13	To identify pass/fail/etc-history of test executions, the test results database should be processed.	The data collected is stored and managed.
W_DR_14	To identify human-identified risks and risk levels, the risk management data in spreadsheets could be processed.	The data collected is stored and managed.
W_DR_15	To identify the topology of a test system, which may be useful in a bigger analysis, the test system topology descriptions could be used.	The data collected is stored and managed.
W_DR_16	To identify the topology of test cases, which may be useful in a bigger analysis, the test case topology descriptions could be used.	The data collected is stored and managed.

Table 48 Use Case Data Requirements Mapping to the "Storage Capabilities" Component

Requirement ID	Requirement Description	Rationale
VCE_R05	Customise standards based modelling frameworks (e.g. UAF, SysML, UML) and metamodels to develop system, software, data architecture models	The interface can enable the storage of the models adhering to the developed frameworks.
VCE_R07	Development of standard data classification, reusable definition, representation, usage	The interface can enable the storage of the related data.
W_R_2	Quality monitoring and predictions in devops process	The data collected at runtime needs to be stored.
W_R_3	Extract data from steps into DevOps process.	The data collected from the DevOps process needs to be stored.
W_R_4	Log file storing, indexing, searching, clustering and comparing	The collected log files need to be stored.
BT_R01	NLP contextual analysis of requirements and match against database of responses/solutions	The interface will enable the storage of the data and models.
BT_R02	ML aided control model parameterization during propulsion system testing	The interface will enable the storage of the data and models.

Table 49 Use Case Requirements Mapping to the "IF-INFRA-STORING-ARTEFACTS" Interface

Data Requirement ID	Data Requirement Description	Rationale

AVL_SEC_DR01	CAN data of a realistically behaving vehicular power train to train a plausibility model ANN	The storage component must provide safe and secure storage (authenticity, integrity and availability must be assured).
AVL_SEC_DR02	CAN data of a realistically behaving vehicular power train to train an anomaly detection ANN	The storage component must provide safe and secure storage (authenticity, integrity and availability must be assured).
W_DR_02	To identify non-trivial indicators for quality shortcomings, the test cases could be parsed with NLP.	The collected data needs to be stored.
W_DR_03	To identify non-trivial indicators for quality shortcomings, the source code and recent changes could be parsed.	The collected data needs to be stored.
W_DR_05	To identify non-trivial indicators for quality shortcomings, the test scripts could be parsed.	The collected data needs to be stored.
W_DR_07	To identify non-trivial indicators for quality shortcomings, the logs from static code analysis could be parsed.	The collected data needs to be stored.
W_DR_08	To identify non-trivial indicators for quality shortcomings, the compilation logs could be parsed.	The collected data needs to be stored.
W_DR_09	To identify anomalies or gradual degradation in performance, the test execution logs could be parsed	The collected data needs to be stored.
W_DR_10	To identify anomalies or gradual degradation in performance, as well as functional issues and error messages, the device communication logs should be parsed.	The collected data needs to be stored.
W_DR_11	To extract information on non-functional characteristics (from e.g. free, top, etc.), the device communication logs should be processed.	The collected data needs to be stored.
W_DR_12	To identify the current configuration of a device being tested, the device communication logs could be processed.	The collected data needs to be stored.
W_DR_13	To identify pass/fail/etc-history of test executions, the test results database should be processed.	The collected data needs to be stored.
W_DR_14	To identify human-identified risks and risk levels, the risk management data in spreadsheets could be processed.	The collected data needs to be stored.
W_DR_15	To identify the topology of a test system, which may be useful in a bigger analysis, the test system topology descriptions could be used.	The collected data needs to be stored.

W_DR_16	To identify the topology of test cases, which may be useful in a bigger analysis, the test case topology descriptions could be used.	The collected data needs to be stored.

Table 50 Use Case Data Requirements Mapping to the "IF-INFRA-STORING-ARTEFACTS" Interface

Requirement ID	Requirement Description	Rationale
VCE_R05	Customise standards based modelling frameworks (e.g. UAF, SysML, UML) and metamodels to develop system, software, data architecture models	The interface can enable the retrieval of models adhering to the developed frameworks.
VCE_R07	Development of standard data classification, reusable definition, representation, usage	The interface can enable the retrieval of the related data.
W_R_2	Quality monitoring and predictions in devops process	Once stored, the data collected at runtime needs to be retrieved.
W_R_3	Extract data from steps into DevOps process.	Once stored, the data collected from the DevOps process needs to be retrieved.
W_R_4	Log file storing, indexing, searching, clustering and comparing	Once stored, the log files collected at runtime need to be retrieved.
BT_R01	NLP contextual analysis of requirements and match against database of responses/solutions	The interface will enable retrieving both the intermediate data that have been used to train the model and the evaluation data.
BT_R02	ML aided control model parameterization during propulsion system testing	The interface will enable retrieving both the intermediate data that have been used to train the models and the evaluation data.

Table 51 Use Case Requirements Mapping to the "IF-INFRA-RETRIEVING-ARTEFACTS" Interface

Data Requirement ID	Data Requirement Description	Rationale
W_DR_02	To identify non-trivial indicators for quality shortcomings, the test cases could be parsed with NLP.	The collected data must be retrieved.
W_DR_03	To identify non-trivial indicators for quality shortcomings, the source code and recent changes could be parsed.	The collected data must be retrieved.
W_DR_05	To identify non-trivial indicators for quality shortcomings, the test scripts could be parsed.	The collected data must be retrieved.
W_DR_07	To identify non-trivial indicators for quality shortcomings, the logs from static code analysis could be parsed.	The collected data must be retrieved.

W_DR_08	To identify non-trivial indicators for quality shortcomings, the compilation logs could be parsed.	The collected data must be retrieved.
W_DR_09	To identify anomalies or gradual degradation in performance, the test execution logs could be parsed	The collected data must be retrieved.
W_DR_10	To identify anomalies or gradual degradation in performance, as well as functional issues and error messages, the device communication logs should be parsed.	The collected data must be retrieved.
W_DR_11	To extract information on non-functional characteristics (from e.g. free, top, etc.), the device communication logs should be processed.	The collected data must be retrieved.
W_DR_12	To identify the current configuration of a device being tested, the device communication logs could be processed.	The collected data must be retrieved.
W_DR_13	To identify pass/fail/etc-history of test executions, the test results database should be processed.	The collected data must be retrieved.
W_DR_14	To identify human-identified risks and risk levels, the risk management data in spreadsheets could be processed.	The collected data must be retrieved.
W_DR_15	To identify the topology of a test system, which may be useful in a bigger analysis, the test system topology descriptions could be used.	The collected data must be retrieved.
W_DR_16	To identify the topology of test cases, which may be useful in a bigger analysis, the test case topology descriptions could be used.	The collected data must be retrieved.

Table 52 Use Case Data Requirements Mapping to the "IF-INFRA-RETRIEVING-ARTEFACTS" Interface

Several Use Case Providers have explicitly expressed specific needs in terms of storage capabilities. For these partners in particular, data and model storage is a key concern in the context of their respective use cases. However, this does not mean that the other Use Case Providers are not using such storage capabilities. Indeed, data and model storage are considered as a default requirement in the context of the AIDOaRt Core Tool Set.

A couple of Use Case Providers have explicitly expressed specific data needs in terms of storage capabilities. For these partners in particular, a focus is made on the problem of data storage in the context of their respective use case. However, this does not mean that the other Use Case Providers do not have constraints nor objectives in terms of data storage. As a consequence, we have been monitoring the data storage issues in the general case in the context of the AIDOaRt Core Tool Set.

4.2 Mapping to Computation Capabilities

In this section, we present the mapping of the use case requirements and data requirements to the "Computation Capabilities" component of the Core Tool Set.

Requirement ID	Requirement Description	Rationale
BT_R02	ML aided control model parameterization during propulsion system testing	Alstom will use the Computation Capabilities Component to train models and predict the parameters.
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	Enough computing power for model learning must be present.
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	Enough computing power for model learning must be present.
AVL_SEC_R04	Use formal model checking methods to derive test cases out of a system model	Enough computing power for model checking must be present.
AVL_SEC_R05	Use AI (ML) methods to learn on the normal behaviour on a powertrain CAN	Enough computing power for model learning must be present.
AVL_SEC_R06	Use AI (ML) methods to learn detect abnormal behaviour on a CAN	Enough computing power for model learning must be present.
W_R_1	AI/ML-powered monitoring/automation of devops process	Data Analysis is required for monitoring or automation of the DevOps process.
W_R_2	Quality monitoring and predictions in devops process	Data Analysis is required for monitoring or automation of the DevOps process.
W_R_4	Log file storing, indexing, searching, clustering and comparing	In order to analyse log files, e.g., to cluster them, some computations are needed.
AVL_SEC_R08	Use live connection to remotely transfer CAN messages	Resources for secure transport must be present (e.g. HSM).
AVL_SEC_R09	Use secure remote transfer connection	Resources for secure transport must be present (e.g. HSM).
PRO_R05	Detect automatically anomalies in the solution during the execution based on AI	Sufficient computation capabilities are needed to perform the AI-based execution/detection.
PRO_R06	Detect and predict high/low resources demand based on AI	Sufficient computation capabilities are needed to perform the AI-based detection/prediction.
PRO_R07	Monitor the platform in real time to reduce the downtime and the data lost	Sufficient computation capabilities are needed to perform the runtime monitoring.
PRO_R08	Self-healing and self-learning solution to minimise the downtime of the platform by detecting and correcting the problems automatically. Avoiding the manual recovery	Sufficient computation capabilities are needed to perform the automated healing/learning.

	of the problems

Table 53 Use Case Requirements Mapping to the "Computation Capabilities" Component

Requirement ID	Requirement Description	Rationale
W_R_4	Log file storing, indexing, searching, clustering and comparing	Log files could be input for various computational algorithms, e.g. for clustering, indexing, searching or comparing one log file with another.
BT_R02	ML aided control model parameterization during propulsion system testing	The interface will enable training and evaluating different forms of models.
PRO_R07	Monitor the platform in real time to reduce the downtime and the data lost	Uses INFRA-COMPUTING-FROM-ARTEFACTS to produce data related with the status of the platform.
W_R_1	AI/ML-powered monitoring/automation of devops process	Monitoring and automation may rely on computed analyses from artefacts.
W_R_2	Quality monitoring and predictions in devops process	Quality monitoring may rely on computed analyses from artefacts.

Table 54 Use Case Requirements Mapping to the "IF-INFRA-COMPUTING-FROM-ARTEFACTS" Interface

Requirement ID	Requirement Description	Rationale
PRO_R05	Detect automatically anomalies in the solution during the execution based on AI	Uses INFRA-GETTING-COMPUTATION-RESULTS to obtain data produced by the monitoring platform.
PRO_R06	Detect and predict high/low resources demand based on AI	Uses INFRA-GETTING-COMPUTATION-RESULTS to obtain data produced by the monitoring platform.
PRO_R08	Self-healing and self-learning solution to minimise the downtime of the platform by detecting and correcting the problems automatically. Avoiding the manual recovery of the problems	Uses INFRA-GETTING-COMPUTATION-RESULTS to obtain data produced by the monitoring platform and anomaly detection functionality.
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	Enough computing power for model learning must be present.
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	Enough computing power for model learning must be present.
AVL_SEC_R04	Use formal model checking methods to derive test cases out of a system model	Enough computing power for model learning must be present.

AVL_SEC_R05	Use AI (ML) methods to learn on the normal behaviour on a powertrain CAN	Enough computing power for model learning must be present.
AVL_SEC_R06	Use AI (ML) methods to learn detect abnormal behaviour on a CAN	Enough computing power for model learning must be present.
AVL_SEC_R08	Use live connection to remotely transfer CAN messages	Resources for secure transport must be present (e.g. HSM)
AVL_SEC_R09	Use secure remote transfer connection	Resources for secure transport must be present (e.g. HSM)

Table 55 Use Case Requirements Mapping to the "IF-INFRA-GETTING-COMPUTATION-RESULTS" Interface

Several Use Case Providers have explicitly expressed specific needs in terms of computation capabilities. For these partners in particular, the capacity to perform complex computations in a scalable way is a key concern in the context of their respective use cases (notably because of the intensive use of AI/ML in these use cases, cf. the WP4 deliverables). However, this does not mean that the other Use Case Providers do not use such computation capabilities when deploying parts of the AIDOaRt solution on their own scenarios. Indeed, the use of AI/ML in the context of AIDOaRt will require high-level capabilities in terms of computation. As a consequence, we paid a particular attention to these capabilities in the context of the AIDOaRt Core Tool Set.

4.3 Mapping to Data Handling Capabilities

In this section, we present the mapping of the use case requirements and data requirements to the "Data Handling Capabilities" component of the Core Tool Set.

Requirement ID	Requirement Description	Rationale
VCE_R05	Customise standards based modelling frameworks (e.g. UAF, SysML, UML) and metamodels to develop system, software, data architecture models	Data handling component is expected to ensure the customised frameworks are imported, edited and exported in a reusable and interoperable fashion. VCE would use data handling to allow for reuse and interoperability between third party tools.
VCE_R07	Development of standard data classification, reusable definition, representation, usage	Data handling component is expected to ensure the models and artefacts used in development are saved in interoperable formats for reuse. VCE would use data handling to allow for further reuse and interchange with third part tools.
BT_R01	NLP contextual analysis of requirements and match against database of responses/solutions	The Data Handling Component is expected to ensure that customised requirements are imported, processed and exported in a reusable

		and interoperable fashion. Alstom would use data handling to allow for reuse and interoperability between third-party tools.
BT_R02	ML aided control model parameterization during propulsion system testing	Alstom will use the Data Handling Capabilities Component to select the data that will be used in training and testing the parametrization model.
AVL_SEC_R01	Use automata learning and ML techniques to derive SUT models	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R04	Use formal model checking methods to derive test cases out of a system model	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R05	Use AI (ML) methods to learn on the normal behaviour on a powertrain CAN	The data handling should ensure that the machine model is compatible with the fuzzer.
AVL_SEC_R06	Use AI (ML) methods to learn detect abnormal behaviour on a CAN	The data handling should ensure that the machine model is compatible with the fuzzer.
AVL_SEC_R07	Use intelligent fuzzing techniques on a CAN bus	The data handling should ensure that the machine model is compatible with the fuzzer.
W_R_4	Log file storing, indexing, searching, clustering and comparing	The large amounts of log data from the DevOps process requires more than trivial ingestion, handling and storage in order to support e.g. search and retrieval of log files and logged data.
AVL_RDE_R03	Automated multi-source data analysis of the real driving test data such that the relevant features of the driver behaviour can be clustered (e.g. highway driving, low speed driving, cornering, braking, acceleration,…). To be used for understanding the driving conditions.	Identifying the data features of the observed driver behaviour relevant for the specific traffic situation

Table 56 Use Case Requirements Mapping to the "Data Handling Capabilities" Component

Data Requirement ID	Data Requirement Description	Rationale
PRO_log	All the resources of the platform including the IoT devices installed in the vehicles will provide information about resource usage (Memory, CPU, Disk).	Uses IF-DATA-SAVING to keep data from the sensors.
PRO_IoT	IoT devices periodically send data collected by the different sensors they contain. This data is sent via JSON messages. Regarding trucks and cranes, they send one message per second with information about the vehicle's operation/status. The important thing about this data is to verify that it is sent and that the messages are not lost. The content of the messages is not relevant to the purpose of the use case.	Uses IF-DATA-SAVING to keep data from the sensors.
PRO_Monitoring	The monitoring platform with collaboration with some AI algorithms will detect problems in the platform. Every time that a problem is found an alarm/notification will be generated.	Uses IF-DATA-SAVING to keep data from the platform.

Table 57 Use Case Data Requirements Mapping to the "Data Handling Capabilities" Component

Requirement ID	Requirement Description	Rationale
VCE_R05	Customise standards based modelling frameworks (e.g. UAF, SysML, UML) and metamodels to develop system, software, data architecture models	The interface can enable the loading of data or models from standard formats across various tools that adhere to the VCE customised framework.
VCE_R07	Development of standard data classification, reusable definition, representation, usage	The interface can enable the loading of the related data.
BT_R01	NLP contextual analysis of requirements and match against database of responses/solutions	The interface will enable the loading of the requirements from requirements management software such as Doors.
BT_R02	ML aided control model parameterization during propulsion system testing	The interface will enable the loading of the motor operation data from Excel files.
AVL_SEC_R01	Use automata learning and ML techniques to derive SUT models	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.

AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R04	Use formal model checking methods to derive test cases out of a system model	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R05	Use AI (ML) methods to learn on the normal behaviour on a powertrain CAN	The data handling should ensure that the machine model is compatible with the fuzzer.
AVL_SEC_R06	Use AI (ML) methods to learn detect abnormal behaviour on a CAN	The data handling should ensure that the machine model is compatible with the fuzzer.
AVL_SEC_R07	Use intelligent fuzzing techniques on a CAN bus	The data handling should ensure that the machine model is compatible with the fuzzer.

Table 58 Use Case Requirements Mapping to the "IF-DATA-LOADING" Interface

Data Requirement ID	Data Requirement Description	Rationale
PRO_IoT	IoT devices periodically send data collected by the different sensors they contain. This data is sent via JSON messages. Regarding trucks and cranes, they send one message per second with information about the vehicle's operation/status. The important thing about this data is to verify that it is sent and that the messages are not lost. The content of the messages is not relevant to the purpose of the use case.	Uses IF-DATA-LOADING to obtain data coming from IoT devices.
PRO_log	All the resources of the platform including the IoT devices installed in the vehicles will provide information about resource usage (Memory, CPU, Disk).	Uses all data management interfaces to handle the data from devices. Uses IF-DATA-LOADING to load data coming from IoT devices linked to the SPMP platform.
PRO_Monitoring	The monitoring platform with collaboration with some AI algorithms will detect problems in the platform. Every time that a problem is found an alarm/notification will be generated.	Uses IF-DATA-LOADING to load data coming from IoT devices linked to the SPMP platform.

Table 59 Use Case Data Requirements Mapping to the "IF-DATA-LOADING" Interface

Requirement ID	Requirement Description	Rationale

BT_R01	NLP contextual analysis of requirements and match against database of responses/solutions	The interface will enable the filtering/selection of the requirements data to prepare proper data for the next processing step.
BT_R02	ML aided control model parameterization during propulsion system testing	The interface will enable the filtering/selection of the motor operation data to prepare proper data for the next processing step.

Table 60 Use Case Requirements Mapping to the "IF-DATA-NAVIGATION" Interface

Data Requirement ID	Data Requirement Description	Rationale
PRO_IoT	IoT devices periodically send data collected by the different sensors they contain. This data is sent via JSON messages. Regarding trucks and cranes, they send one message per second with information about the vehicle's operation/status. The important thing about this data is to verify that it is sent and that the messages are not lost. The content of the messages is not relevant to the purpose of the use case.	Uses IF-DATA-NAVIGATION to map through data coming from IoT devices in a comfortable way.
PRO_log	All the resources of the platform including the IoT devices installed in the vehicles will provide information about resource usage (Memory, CPU, Disk).	Uses IF-DATA-NAVIGATION to navigate through log_data coming from SPMP platform.
PRO_Monitoring	The monitoring platform with collaboration with some AI algorithms will detect problems in the platform. Every time that a problem is found an alarm/notification will be generated.	Uses IF-DATA-NAVIGATION to navigate through data coming from sensors.

Table 61 Use Case Data Requirements Mapping to the "IF-DATA-NAVIGATION" Interface

Requirement ID	Requirement Description	Rationale
W_R_4	Log file storing, indexing, searching, clustering and comparing	In order to search in log files after a string, e.g. an uncommon error message, one must be able to query a log repository.
BT_R01	NLP contextual analysis of requirements and match against database of responses/solutions	The interface will enable querying the new requirements data as input to the developed models.
BT_R02	ML aided control model parameterization during propulsion system testing	The interface will enable querying the new motor operation data as input to the developed models.

AVL_RDE_R03	Automated multi-source data analysis of the real driving test data such that the relevant features of the driver behaviour can be clustered (e.g. highway driving, low speed driving, cornering, braking, acceleration,…). To be used for understanding the driving conditions.	Identifying the data features of the observed driver behaviour relevant for the specific traffic situation

Table 62 Use Case Requirements Mapping to the "IF-DATA-QUERYING" Interface

Data Requirement ID	Data Requirement Description	Rationale
PRO_IoT	IoT devices periodically send data collected by the different sensors they contain. This data is sent via JSON messages. Regarding trucks and cranes, they send one message per second with information about the vehicle's operation/status. The important thing about this data is to verify that it is sent and that the messages are not lost. The content of the messages is not relevant to the purpose of the use case.	Uses IF-DATA-QUERYING to request data from IoT devices.
PRO_Monitoring	The monitoring platform with collaboration with some AI algorithms will detect problems in the platform. Every time that a problem is found an alarm/notification will be generated.	Uses IF-DATA-QUERYING to request data from platform.
PRO_log	All the resources of the platform including the IoT devices installed in the vehicles will provide information about resource usage (Memory, CPU, Disk).	Uses IF-DATA-QUERYING to request data from sensors.

Table 63 Use Case Data Requirements Mapping to the "IF-DATA-QUERYING" Interface

Requirement ID	Requirement Description	Rationale
VCE_R05	Customise standards based modelling frameworks (e.g. UAF, SysML, UML) and metamodels to develop system, software, data architecture models	The interface can enable the saving of data or models from standard formats across various tools that adhere to the VCE customised framework.
VCE_R07	Development of standard data classification, reusable definition, representation, usage	The interface can enable the saving of the related data.
AVL_SEC_R07	Use intelligent fuzzing techniques on a CAN bus	The data handling should ensure that the machine model is compatible with the fuzzer.
AVL_SEC_R06	Use AI (ML) methods to learn detect abnormal behaviour on a CAN	The data handling should ensure that the machine model is compatible with the fuzzer.

AVL_SEC_R05	Use AI (ML) methods to learn on the normal behaviour on a powertrain CAN	The data handling should ensure that the machine model is compatible with the fuzzer.
AVL_SEC_R04	Use formal model checking methods to derive test cases out of a system model	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.
AVL_SEC_R01	Use automata learning and ML techniques to derive SUT models	The data handling should ensure that the automata learning model is compatible with plausibility and model checking.

Table 64 Use Case Requirements Mapping to the "IF-DATA-SAVING" Interface

Data Requirement ID	Data Requirement Description	Rationale
PRO_IoT	IoT devices periodically send data collected by the different sensors they contain. This data is sent via JSON messages. Regarding trucks and cranes, they send one message per second with information about the vehicle's operation/status. The important thing about this data is to verify that it is sent and that the messages are not lost. The content of the messages is not relevant to the purpose of the use case.	Uses IF-DATA-SAVING to keep data from IoT devices.
PRO_Monitoring	The monitoring platform with collaboration with some AI algorithms will detect problems in the platform. Every time that a problem is found an alarm/notification will be generated.	Uses IF-DATA-SAVING to keep data from SPPM platform.
PRO_log	All the resources of the platform including the IoT devices installed in the vehicles will provide information about resource usage (Memory, CPU, Disk).	Uses IF-DATA-SAVING to keep data from sensors.

Table 65 Use Case Data Requirements Mapping to the "IF-DATA-SAVING" Interface

Different Use Case Providers have explicitly expressed specific needs in terms of data handling capabilities. For these partners in particular, the capacity to handle data in a scalable way is a key concern in the context of their respective use cases. This is notably because of the intensive use of AI/ML in these use cases (cf. WP4) and/or of the use of very large data sets such as logs (cf. WP2). Moreover, we observe a partial correlation with the mapping to Storage and Computation capabilities as presented before in this deliverable. However, this does not mean that the other Use Case Providers do not use such data handling capabilities when deploying parts of the AIDOaRt solution on their own scenarios. Please refer to the Data Engineering Tool Set for more details on these data aspects and dedicated support in the context of the project.

4.4 Mapping to Model-Based Capabilities

In this section, we present the mapping of the use case requirements and data requirements to the "Model-Based Capabilities" component of the Core Tool Set.

Requirement ID	Requirement Description	Rationale
VCE_R05	Customise standards based modelling frameworks (e.g. UAF, SysML, UML) and metamodels to develop system, software, data architecture models	The model-based components are expected to ensure that model-based development and engineering is possible. All capabilities of these components are expected to be extended to customizable modelling frameworks and standards to allow a traceable and interoperable modelling approach. VCE would use the model-based component to enable a traceable and interoperable workflow for model development and management during design of system, software, data and architecture models.
VCE_R07	Development of standard data classification, reusable definition, representation, usage	The model-based components are expected to ensure that fully model based capabilities are available when dealing with standardised models. VCE would use the model-based component so developers and engineers can utilise required MBSE capabilities.
AVL_TCR_R04	Deploy any data driven model generated with state-of-the-art frameworks to platform independent C++ code.	The Model-Based Capabilities component is expected to ensure the capability to store models in different types of serialisation or representation formats. AVL would Model-Based Capabilities solution to deploy data driven models to platform independent C++ code.

BT_R02	ML aided control model parameterization during propulsion system testing	Alstom will use the Model Handling Capabilities Component to work with the temperature model and the parameters' prediction model.
CSY_R03	Use classification on project PO to predict the best tool for automatic proving	CSY can use Model Based capabilities to classify models of proof obligations based upon the tools that can solve them.
CSY_R04	Use deep learning to write abstraction of implementation	CSY can use Model-based capabilities, and particularly model transformation to generate abstraction of B-Implementation models
CSY_R05	Use deep learning to write refinement of specification	CSY can use Model-based capabilities, and particularly model transformation to generate refinement of B-specification models.
CSY_R02	Use reinforcement and deep learning techniques on proof theory and solving	CSY can use Model based capabilities to navigate in banks of proof demonstration and to select and adapt demonstrations to new proof obligations.
AVL_SEC_R04	Use formal model checking methods to derive test cases out of a system model	The Model-Based Capabilities component is expected to the model format to be transferred into a format that could be used for formal verification.
AVL_TCR_R02	Implement approaches of data driven models based on AVL provided testbed data with the aim to do simulation.	Transform the learned model to fulfil the requirements of an executable model within an simulation environment
AVL_TCR_R03	Finding new AI/ML technologies to increase and extend the model efficiency and capabilities of the current modelling solutions.	PUMA ML Framework is capable of transforming the initial model in order to adapt to a new user requirement
AVL_MBT_R01	Toolkit to generate test cases based on an Unified Modelling Language (UML) model in order to make initial measurements to generate a surrogate model for large (dynamic and or combinatorial) systems. For example, uniform or Sobol sampling for continuous regression models or Amplitude Modulated Pseudo-Random Bit Sequences (APRBS) for dynamic models.	Initial model should be adapted to the specific user requirements for test case generation

AVL_MBT_R02	An ML toolkit to generate surrogate models for system structures with a lot of binary or combinatorial inputs. Particularly we want to focus on large system structures that can be partitioned into several subsystems which can be modelled independently.	Combine/merge multiple models with specific aspects into a single one for a specific purpose
AVL_MBT_R03	An AI method to generate new test cases based on 1) calibration demands 2) information from previous measurements 3) the ML model from MBT_R_2. New test cases should be generated in order to improve model quality of the ML model, while avoiding similar measurements throughout the procedure and fulfilling calibration demands, for example given by constraints.	Based on the previous testing recordings/results, model adaptations should be generated to optimise the next testing iteration with improved model characteristics
AVL_TCV_R02	The verdict given by the SCENIUS Test Case Selection Validator must be understandable/explainable by humans	Provide an view to the user that allows him to judge about the model performance
AVL_TCV_R03	The new parameter values given by the SCENIUS parameter recommender must lead to critical situations which are not covered by the generated Tests from the SCENIUS test case generator.	Show counter-examples derived from the model regarding its expected behaviour
AVL_ODP_R01	Implement approaches of learning data-driven models based on project data (especially Key Performance Index (KPI) and parameter value evolution) provided by AVL. The models are utilised to compute the maturity of a specific KPI result in a specific project at a specific point in time in a standardised and objective way.	Provide traceability to models of previous projects. Derive a view of the previous model to judge certain KPI (such as model maturity level).
AVL_ODP_R02	Implement approaches of data-driven models based on project data (especially Key Performance Index (KPI) and parameter value evolution) provided by AVL. The models are utilised to forecast the KPI (and parameter) value evolution in the project.	Provide traceability to previous models (of previous projects) and their specific KPIs at a certain point of the DevOps chain to conclude the actual KPI (i.e. model matureness) of the actual model in the actual project.
AVL_ODP_R03	Implement approaches of data-driven models based on project data (especially Key Performance Index (KPI) and parameter value evolution) provided by AVL. The models are utilised to assess the information gain for specific experiments in order to identify experiments that provide little information gain and can thus be e.g. skipped in future projects.	Provide another view on the actual model containing recommender information derived from the evaluated KPI (e.g. model matureness) to optimise the ongoing DevOps process (such as skipping test with little expected benefits)

AVL_SEC_R01	Use automata learning and ML techniques to derive SUT models	Automata Learning is an active technique that queries systems to infer a model
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	Querying for plausibility checking
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	Querying for ANN training

Table 66 Use Case Requirements Mapping to the "Model-Based Capabilities" Component

Requirement ID	Requirement Description	Rationale
CSY_R03	Use classification on project PO to predict the best tool for automatic proving	The interface should permit loading xml POG models to properly classify in the right prover category.
CSY_R02	Use reinforcement and deep learning techniques on proof theory and solving	The interface should permit loading xml POG models all along the interactive proving process to properly run or assist the proof.

Table 67 Use Case Requirements Mapping to the "IF-MODEL-LOADING" Interface

Requirement ID	Requirement Description	Rationale
AVL_SEC_R01	Use automata learning and ML techniques to derive SUT models	Automata Learning is an active technique that queries systems to infer a model
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	Querying for plausibility checking
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	Querying for ANN training
BT_R02	ML aided control model parameterization during propulsion system testing	Quarrying for model training, validating, and run-time usage.

Table 68 Use Case Requirements Mapping to the "IF-MODEL-QUERYING" Interface

Requirement ID	Requirement Description	Rationale
AVL_TCR_R02	Implement approaches of data driven models based on AVL provided testbed data with the aim to do simulation.	Transform the learned model to fulfil the requirements of an executable model within an simulation environment
AVL_TCR_R03	Finding new AI/ML technologies to increase and extend the model efficiency and capabilities of the current modelling solutions.	PUMA ML Framework is capable of transforming the initial model in order to adapt to a new user requirement
AVL_MBT_R01	Toolkit to generate test cases based on an Unified Modelling Language (UML) model in order to make initial measurements to	Initial model should be adapted to the specific user

	generate a surrogate model for large (dynamic and or combinatorial) systems. For example, uniform or Sobol sampling for continuous regression models or Amplitude Modulated Pseudo-Random Bit Sequences (APRBS) for dynamic models.	requirements for test case generation
AVL_MBT_R02	An ML toolkit to generate surrogate models for system structures with a lot of binary or combinatorial inputs. Particularly we want to focus on large system structures that can be partitioned into several subsystems which can be modelled independently.	Combine/merge multiple models with specific aspects into a single one for a specific purpose
AVL_MBT_R03	An AI method to generate new test cases based on 1) calibration demands 2) information from previous measurements 3) the ML model from MBT_R_2. New test cases should be generated in order to improve model quality of the ML model, while avoiding similar measurements throughout the procedure and fulfilling calibration demands, for example given by constraints.	Based on the previous testing recordings/results, model adaptations should be generated to optimise the next testing iteration with improved model characteristics
AVL_SEC_R04	Use formal model checking methods to derive test cases out of a system model	The Model-Based Capabilities component is expected to the model format to be transferred into a format that could be used for formal verification.

Table 69 Use Case Requirements Mapping to the "IF-MODEL-TRANSFORMATION" Interface

Requirement ID	Requirement Description	Rationale
AVL_TCV_R02	The verdict given by the SCENIUS Test Case Selection Validator must be understandable/explainable by humans	Provide an view to the user that allows him to judge about the model performance
AVL_TCV_R03	The new parameter values given by the SCENIUS parameter recommender must lead to critical situations which are not covered by the generated Tests from the SCENIUS test case generator.	Show counter-examples derived from the model regarding its expected behaviour
AVL_ODP_R01	Implement approaches of learning data-driven models based on project data (especially Key Performance Index (KPI) and parameter value evolution) provided by AVL. The models are utilised to compute the maturity of a specific KPI result in a specific project at a specific point in time in a standardised and objective way.	Provide traceability to models of previous projects. Derive a view of the previous model to judge certain KPI (such as model maturity level).
AVL_ODP_R02	Implement approaches of data-driven models based on project data (especially Key Performance Index (KPI) and parameter value	Provide traceability to previous models (of previous projects) and their

	evolution) provided by AVL. The models are utilised to forecast the KPI (and parameter) value evolution in the project.	specific KPIs at a certain point of the DevOps chain to conclude the actual KPI (i.e. model matureness) of the actual model in the actual project.
AVL_ODP_R03	Implement approaches of data-driven models based on project data (especially Key Performance Index (KPI) and parameter value evolution) provided by AVL. The models are utilised to assess the information gain for specific experiments in order to identify experiments that provide little information gain and can thus be e.g. skipped in future projects.	Provide another view on the actual model containing recommender information derived from the evaluated KPI (e.g. model matureness) to optimise the ongoing DevOps process (such as skipping test with little expected benefits)

Table 70 Use Case Requirements Mapping to the "IF-MODEL-TRACEABILITY-FEDERATION" Interface

Requirement ID	Requirement Description	Rationale
CSY_R04	Use deep learning to write abstraction of implementation	This interface would be used to generate abstractions from implementation. It can serve to derive abstract variables, infer their types, and discover general properties on those variables. Ideally, it could also serve to write contracts, i.e. pre- and post-conditions, of functions.
CSY_R05	Use deep learning to write refinement of specification	This interface would be used to generate implementation from abstractions. It can serve to derive concrete variables, infer their types, and write block of code that follow general patterns such as initializations or loops.
AVL_TCR_R04	Deploy any data driven model generated with state-of-the-art frameworks to platform independent C++ code.	Deploy any data driven model generated with state-of-the-art frameworks to platform independent C++ code.

Table 71 Use Case Requirements Mapping to the "IF-MODEL-CODE-GENERATION" Interface

Requirement ID	Requirement Description	Rationale
VCE_R05	Customise standards based modelling frameworks (e.g. UAF, SysML, UML) and metamodels to develop system, software, data architecture models	The interface can enable traceability between various artefacts developed as part of the customised frameworks towards various standards and requirements.
VCE_R07	Development of standard data classification, reusable definition, representation, usage	The interface can enable traceability between artefacts related to the standardised classifications.

Table 72 Use Case Requirements Mapping to the "Modelling Process Tracing" Interface

A significant number of Use Case Providers have explicitly expressed specific needs in terms of model-based capabilities. For these partners in particular, the capacity to create and handle models in a scalable way is a key concern in the context of their respective use cases (notably because of the intensive use of large sets of heterogeneous design time and runtime models). Moreover, we can also see here a partial correlation with the mapping to the previously described Storage, Computation and Data-Handling capabilities. However, this does not mean that the other Use Case Providers do not use such model-based capabilities when deploying parts of the AIDOaRt solution on their own scenarios. Indeed, the AIDOaRt overall solution being model-based, model-based capabilities are fundamental.

4.5 Mapping to Accountability

In this section, we present the mapping of the use case requirements and data requirements to the "Accountability" component of the Core Tool Set.

Requirement ID	Requirement Description	Rationale
AVL_SEC_R08	Use live connection to remotely transfer CAN messages	The accountability component has to ensure that any data transfer operation is replicable.
AVL_SEC_R09	Use secure remote transfer connection	The accountability component has to ensure that any data transfer operation is replicable.
HIB_R03	The AIDOaRT solution must be able to analyse the continuous integration process and detect anomalies.	Accountability is critical for software updates in our Use Case. A clear trace of all of the processes undertaken in the update needs to be present always.
HIB_R04	The AIDOaRT AI algorithms will enable analysing the success of deploying a new version of the POS application.	Accountability is a critical requirement in the Deployment process described in HIB_R04. For all operations during the deployment in all devices, clear proof of action must be produced.

Table 73 Use Case Requirements Mapping to the "Accountability" Component

Data Requirement ID	Data Requirement Description	Rationale
AVL_SEC_DR03	Live (remote) access to CAN IVN	The accountability component has to ensure that any IVN operation is replicable.

Table 74 Use Case Data Requirements Mapping to the "Accountability" Component

Requirement ID	Requirement Description	Rationale
W_R_4	Log file storing, indexing, searching, clustering and comparing	After querying a log repository (e.g. when searching for an uncommon error message), one must be able to get the result somehow.
ABI_R04	Use ML for video elaboration.	ABI_R04 is related to the use of ML for video elaboration. Therefore, it is strictly

		related to the need to collect images and analyse their features/parts towards a model output.
ABI_R11	The system shall detect relevant objects while backing up.	ABI_R011 specifies that the system shall detect relevant objects while backing up. Therefore, it is strictly related to the need to collect images and analyse their features/parts towards a model output.
ABI_R12	The system shall estimate the vehicles approaching speed.	ABI_R12 specifies that the system shall estimate the vehicles approaching speed. Therefore, it is strictly related to the need to collect images and analyse their features/parts towards a model output.
AVL_SEC_R08	Use live connection to remotely transfer CAN messages	The accountability component has to ensure that any data transfer operation is replicable.
AVL_SEC_R09	Use secure remote transfer connection	The accountability component has to ensure that any data transfer operation is replicable.
W_R_1	AI/ML-powered monitoring/automation of devops process	For monitoring and automation in DevOps, Westermo needs to get analysis results.
W_R_2	Quality monitoring and predictions in devops process	For quality monitoring of the DevOps process, Westermo needs to get analysis results.
HIB_R03	The AIDOaRT solution must be able to analyse the continuous integration process and detect anomalies.	We collect the results of the accountability and explainability analysis related to the inspection of the CI/CD process for the generation of TAMUS assets.
HIB_R04	The AIDOaRT AI algorithms will enable analysing the success of deploying a new version of the POS application.	We collect the results of the Accountability and Explainability analyses of the updating process for the TAMUS versions.

Table 75 Use Case Requirements Mapping to the "IF-GENSERV-GETTING-ANALYSIS-RESULTS" Interface

Data Requirement ID	Data Requirement Description	Rationale
AVL_SEC_DR03	Live (remote) access to CAN IVN	The accountability component has to ensure that any IVN operation is replicable.

Table 76 Use Case Data Requirements Mapping to the "IF-GENSERV-GETTING-ANALYSIS-RESULTS" Interface

Requirement ID	Requirement Description	Rationale
ABI_R04	Use ML for video elaboration.	ABI_R04 is related to the use of ML for video elaboration. This concerns the detection of objects, the adopted AI-based models should

		be analysed in terms of explainability.
ABI_R11	The system shall detect relevant objects while backing up.	ABI_R011 specifies that the system shall detect relevant objects while backing up. Therefore, the adopted AI-based models to detect the presence of objects should be analysed in terms of explainability.
ABI_R12	The system shall estimate the vehicles approaching speed.	ABI_R12 specifies that the system shall estimate the vehicles approaching speed. This is strictly related to the detection of tracking of objects. Therefore, the adopted AI-based models to detect the presence of objects should be analysed in terms of explainability.
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	The explainability component has to ensure that the model generation is replicable.
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	The explainability component has to ensure that the model generation is replicable.
AVL_TCV_R02	The verdict given by the SCENIUS Test Case Selection Validator must be understandable/explainable by humans	The verdict given by the SCENIUS Test Case Selection Validator must be understandable/explainable by humans

Table 77 Use Case Requirements Mapping to the "IF-GENSERV-ANALYSING-ARTEFACTS" Interface

Some Use Case Providers have explicitly expressed specific needs in terms of a generic service for accountability. For these partners in particular, accountability is a key concern in the context of their respective use cases. However, this does not mean that the other Use Case Providers do not use at some point such a service offering support for accountability. Indeed, we consider it as a relevant capability when using in conjunction solutions supporting the AI-Augmented Tool Set (cf. WP4).

In addition, a few Use Case Provider explicitly expressed specific data needs in terms of a generic service for accountability. For these partners, a focus is made on the problem of data accountability in the context of their use case. However, this does not mean that the other Use Case Providers do not have constraints nor objectives in terms of data accountability.

4.6 Mapping to Explainability

In this section, we present the mapping of the use case requirements and data requirements to the "Explainability" component of the Core Tool Set.

Requirement ID	Requirement Description	Rationale
AVL_TCV_R02	The verdict given by the SCENIUS Test Case Selection Validator must	The verdict given by the SCENIUS Test Case Selection Validator must be understandable/explainable by humans

	be understandable/explainable by humans	
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	The explainability component has to ensure that the model generation is replicable.
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	The explainability component has to ensure that the model generation is replicable.
W_R_1	AI/ML-powered monitoring/automation of devops process	When an AI makes decisions, then it is very desirable for a human to be able to audit, review and/or understand the rationale for the decision.
W_R_2	Quality monitoring and predictions in devops process	When an AI makes decisions, then it is very desirable for a human to be able to audit, review and/or understand the rationale for the decision.
HIB_R03	The AIDOaRT solution must be able to analyse the continuous integration process and detect anomalies.	When a process of software updates is launched, there needs to be a complete trace of why it was necessary. This needs to motivate the start of the operation.
HIB_R04	The AIDOaRT AI algorithms will enable analysing the success of deploying a new version of the POS application.	Same as in the Update process, any Deploy process requires that proof of the actual motivation of the operation is generated and stored.
ABI_R04	Use ML for video elaboration.	It would be good to get a justification for the AI-supported decision taken during the video elaboration.
ABI_R11	The system shall detect relevant objects while backing up.	It would be good to get a justification for the automatically detected objects.
ABI_R12	The system shall estimate the vehicles approaching speed.	It would be good to get a justification for the automatically estimated speeds.
W_R_4	Log file storing, indexing, searching, clustering and comparing	If log files are clustered, it is desirable to explain the rationale for clusters.

Table 78 Use Case Requirements Mapping to the "Explainability" Component

Requirement ID	Requirement Description	Rationale
W_R_4	Log file storing, indexing, searching, clustering and comparing	After querying a log repository (e.g. when searching for an uncommon error message), one must be able to get the result somehow.
ABI_R04	Use ML for video elaboration.	ABI_R04 is related to the use of ML for video elaboration. Therefore, it is strictly related to the need to collect images and analyse their features/parts towards a model output.
ABI_R11	The system shall detect relevant objects while backing up.	ABI_R011 specifies that the system shall detect relevant objects while backing up.

		Therefore, it is strictly related to the need to collect images and analyse their features/parts towards a model output.
ABI_R12	The system shall estimate the vehicles approaching speed.	ABI_R12 specifies that the system shall estimate the vehicles approaching speed. Therefore, it is strictly related to the need to collect images and analyse their features/parts towards a model output.
AVL_SEC_R08	Use live connection to remotely transfer CAN messages	The accountability component has to ensure that any data transfer operation is replicable.
AVL_SEC_R09	Use secure remote transfer connection	The accountability component has to ensure that any data transfer operation is replicable.
W_R_1	AI/ML-powered monitoring/automation of devops process	For monitoring and automation in DevOps, Westermo needs to get analysis results.
W_R_2	Quality monitoring and predictions in devops process	For quality monitoring of the DevOps process, Westermo needs to get analysis results.
HIB_R03	The AIDOaRT solution must be able to analyse the continuous integration process and detect anomalies.	We collect the results of the accountability and explainability analysis related to the inspection of the CI/CD process for the generation of TAMUS assets.
HIB_R04	The AIDOaRT AI algorithms will enable analysing the success of deploying a new version of the POS application.	We collect the results of the Accountability and Explainability analyses of the updating process for the TAMUS versions.

Table 79 Use Case Requirements Mapping to the "IF-GENSERV-GETTING-ANALYSIS-RESULTS" Interface

Data Requirement ID	Data Requirement Description	Rationale
AVL_SEC_DR03	Live (remote) access to CAN IVN	The accountability component has to ensure that any IVN operation is replicable.

Table 80 Use Case Data Requirements Mapping to the "IF-GENSERV-GETTING-ANALYSIS-RESULTS" Interface

Requirement ID	Requirement Description	Rationale
ABI_R04	Use ML for video elaboration.	ABI_R04 is related to the use of ML for video elaboration. This concerns the detection of objects, the adopted AI-based models should be analysed in terms of explainability.
ABI_R11	The system shall detect relevant objects while backing up.	ABI_R011 specifies that the system shall detect relevant objects while backing up. Therefore, the adopted AI-based models to

		detect the presence of objects should be analysed in terms of explainability.
ABI_R12	The system shall estimate the vehicles approaching speed.	ABI_R12 specifies that the system shall estimate the vehicles approaching speed. This is strictly related to the detection of tracking of objects. Therefore, the adopted AI-based models to detect the presence of objects should be analysed in terms of explainability.
AVL_SEC_R02	Use an ANN to perform plausibility checks on models	The explainability component has to ensure that the model generation is replicable.
AVL_SEC_R03	Train ANN on SUT topology discovery using test observation	The explainability component has to ensure that the model generation is replicable.
AVL_TCV_R02	The verdict given by the SCENIUS Test Case Selection Validator must be understandable/explainable by humans	The verdict given by the SCENIUS Test Case Selection Validator must be understandable/explainable by humans

Table 81 Use Case Requirements Mapping to the "IF-GENSERV-ANALYSING-ARTEFACTS" Interface

Several Use Case Providers explicitly expressed specific needs in terms of a generic service for explainability. For these partners, explainability is a key concern in the context of their respective use cases. However, this does not mean that the other Use Case Providers do not use such a service offering support for explainability. Indeed, we consider it as a relevant capability when using in conjunction solutions supporting the AI-Augmented Tool Set (cf. WP4).

5 Applications of AIDOaRt Core Tool Set Solutions in Use Cases

In order to concretely materialise the final mappings described in the previous sections, this section introduces new recent applications of solutions from the AIDOaRt Core Tool Set that have been applied to address challenges within some of the project's use cases. These use cases are different in terms of scale and kinds of issue to be tackled but are representative of main problems faced when engineering complex CPSs. These examples are practical illustrations of the latest work performed in the context of WP3 associated with the project's objective O2 – "AIDOaRt global model-based framework for the continuous development of CPS".

5.1 VCE Challenge – Model-based CPS Engineering – JKU, UNIVAQ, IMTA, MDU

In the direct continuation of the VCE challenge presented in the previous deliverable D3.3, we pushed further the work and proposed an updated architecture for our model-based solution integrating different tools from the AIDOaRt Core Tool Set.

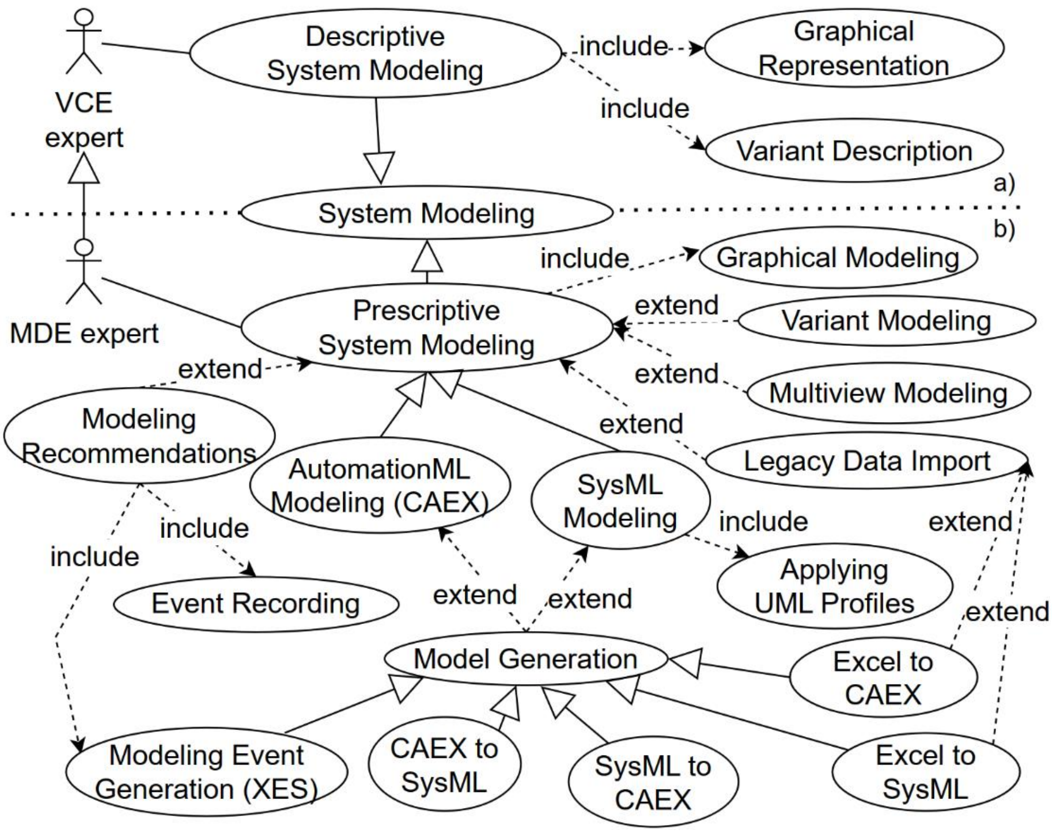

Figure 4 – Actors and use cases in our VCE CPS Model-based Engineering Challenge.

As a summary, Figure 4 shows actors and use cases in the VCE challenges. The use case concerns the need to provide *system modelling* capabilities to VCE engineers. Existing office tools, like Visio and Excel, provide descriptive system modelling capabilities (cf. a) in Figure 4). The current practice at VCE leverages Visio documents for *graphical representation* and Excel sheets for *variant descriptions* documenting product lines with components, their large number of variants, and their different possible combinations. The VCE past experience showed that using industrial-grade APIs provided by office tools often results in complex implementations or plugin customization with poor automation results at the end. In order to overcome this, the AIDOaRt partners proposed to VCE an alternative approach to support *prescriptive system modelling* capabilities in particular (cf. b) in Figure 4).

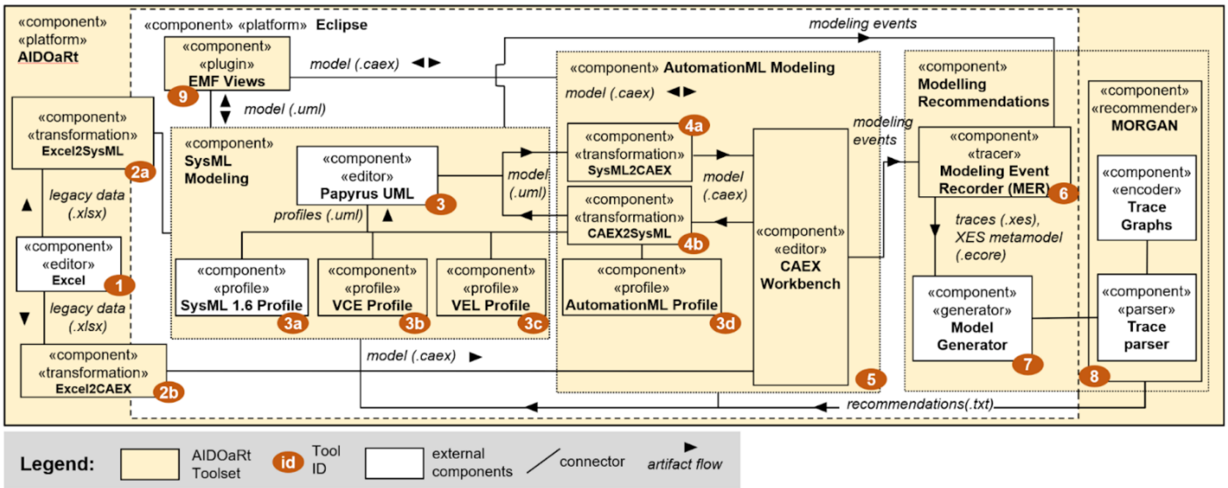

Figure 5 – Solution architecture for our VCE CPS Model-based Engineering Challenge.

Figure 5 depicts a solution architecture detailing the more generic tool landscape sketched in Figure 4 - b. The proposed solution now integrates more AIDOaRt partners' solutions from the AIDOaRt Core Tool Set (i.e. AutomationML Modelling metamodel, EMF Views for model federation, MER for traceability, MORGAN for recommendation) with open-source tools (i.e. Papyrus for design modelling, CAEX for runtime modelling) and new components (e.g. specific model transformations). The objective is to cope more efficiently with the VCE automation and integration needs.

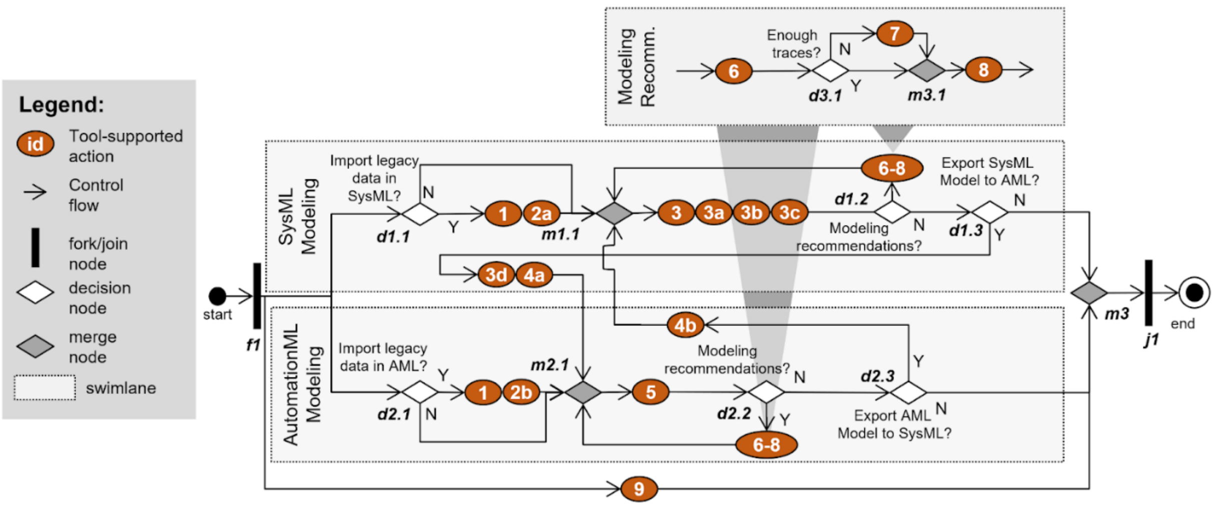

Figure 6 – The engineering workflow enabled by the solution architecture.

Figure 6 depicts the engineering workflow supported by the proposed solution architecture in an activity-like diagram. The components' ids in Figure 5 are used to identify the corresponding supported engineering action in Figure 6. They are grouped into three swim lanes including actions and related tools supporting modelling with i) SysML and ii) AutomationML, and iii) recommendation capabilities.

SysML Modelling

The current solution architecture and engineering workflow concentrate on modelling structural information of VCE systems. For this purpose, SysML supplies block definition and internal block diagrams. For model interoperability reasons this part of the engineering process is currently supported by using the open-source Eclipse Papyrus solution, build atop EMF-compatible technologies. However, other SysML tools could be used as alternative solutions if needed, such as Modelio which is also part of the AIDOaRt Core Tool Set (as we already experienced during the second year of the project).

AutomationML Modelling

AutomationML is a viable alternative to model structural information of VCE systems. The proposed solution architecture allows exchanging engineering information across SysML and AutomationML modelling through model transformations. The modelling in AutomationML is possible through the CAEX Modelling Workbench. The CAEX Modelling Workbench is realised atop EMF-based technologies (EMF, Sirius), which makes it compatible with the other partners' solutions.

Multiview Modelling

As introduced earlier, EMF Views is an approach and corresponding Eclipse/EMF-based tool that provides capabilities for specifying and building views over one or several models that potentially conform to different metamodels. In the context of the VCE use case, EMF Views is used to support Multiview Modelling. It is conceived as an engineering activity independent of SysML and AutomationML modelling. This way, it can also be applied to federate any EMF-based models involved in the CPS engineering process. Currently, it is used to federate SysML and AutomationML models into integrated views that VCE engineers can navigate and query depending on their needs. However, other complementary models can be added to these integrated views in the future if required. Thanks to this, the VCE engineers can more easily get an overall vision of the system under study and make design decisions accordingly without referring to legacy data from Excel sheets.

Modelling Recommendations

With the aim of automatizing the design of CPSs, we provide model recommendations in a Sirius-based environment by combining two tools offered by the solution providers, i.e., MER and MORGAN. The MER plugin captures the modeler's behaviour in terms of operations, i.e., adding, modifying, and deleting classes and attributes. The MER plugin eventually produces a set of traces composed of those events. At this point, MORGAN can be fed with the extracted traces to forecast the next operation giving an initial set. The underpinning algorithm encodes the traces in a graph-based structure and retrieves the most similar operations. In particular, we employ graph kernel similarity that encodes mutual relationships among the traces. We evaluate the capabilities of the two tools by running an initial user study with VCE engineers. Even though the results are encouraging, we plan to extend the evaluation by 1) integrating MER and MORGAN in an Eclipse plugin and 2) involving more participants in the evaluation. Therefore, the AI-related capabilities of the tool will be detailed in the next deliverables of the project in the context of WP4.

5.2 PRO Challenge – Modelling and Monitoring for a Smart Platform (SPMP) – ACO, ITI

The Smart Port Monitoring Platform involves the deployment of a large infrastructure of interconnected elements (CPS) that need to be continuously monitored so that any failure that occurs can be detected and corrected as soon as possible so that the platform does not interrupt its workflow (or only minimally). Therefore, it is necessary to implement systems for anomaly detection and it is also vital to model the infrastructure in advance in order to have the necessary resources available. We have made significant progress in the following areas:

- Anomaly detection techniques;
- Infrastructure modelling to determine the needs of the platform in different scenarios.

Anomaly detection

As previously explained in this deliverable, the multi-level tracing extension for ESDE enables the obtention of traces of significant events at the different layers of an embedded system. It includes relevant software layers (application, RTOS, drivers), and hardware components (e.g. access events at different elements of the memory hierarchy.

This has immediate, traversal advantages for analysis procedures in ACORDE. In effect, it enables, by relying on an open, Licence free setup, to conduct a graphical analysis on the different events and casualties produced by a piece of code. It is illustrated in Figure 7 representation obtained with one of the former examples run. In such an example a main application involves iterative function invocations up to a 4-th nesting level. The analysis of Figure 7 traced signals, corresponding to function invocations & exit events at each nesting level, provides important information about the most active functions, and thus where optimization is worthwhile. Moreover, traces at lower levels, i.e. memory accesses provide important profiling information to understand and obtain the performance of different coding alternatives, or to determine, for instance, memory requirements.

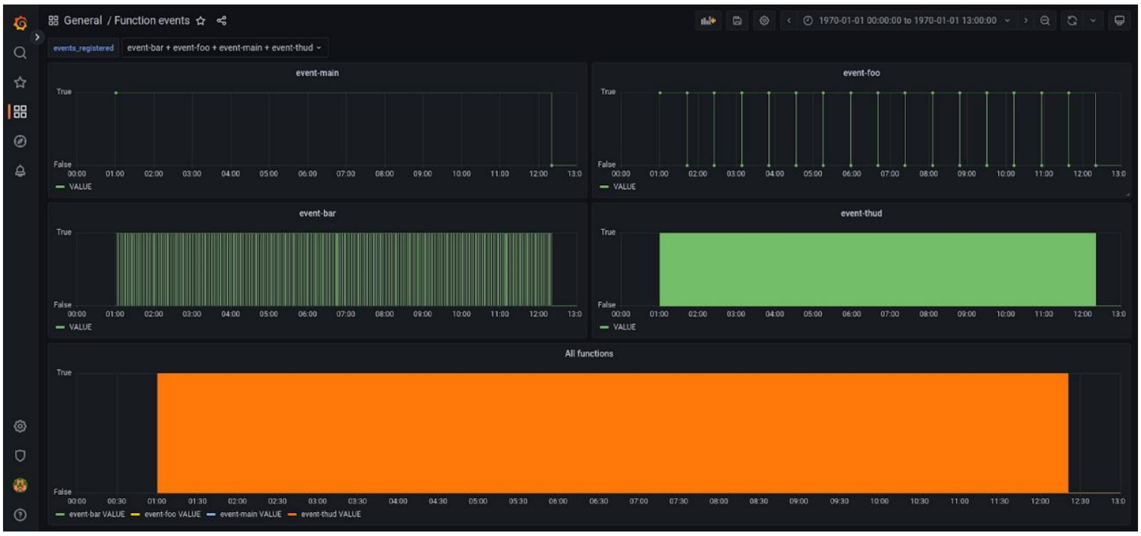

Figure 7 – Enabled trace analysis enables a better understanding of the activity of the different parts of the code, but also of the HW components and thus sharper performance analysis.

Moreover, multi-level tracing is expected to expose the main (or just the existing) dependencies and relations eventually stated by the application and the different levels of the embedded system. The learning of these relations is expected to serve to detect anomalous behaviours (fixed, HW platforms faults), by means of using the same Generic Anomalies Analysis (GAA) techniques currently developed and explored by ACORDE in its application to traces from sensors deployed at the port.

More specifically, the aim of ACORDE is applying GAA on the design of the embedded application and/or determination of the most efficient platform for the positioning IoT developed for the Smart Port Monitoring Platform. For that application, ACORDE is exploring the possibility for a low-cost alternative, which requires the evaluation of the embedded positioning application on top of the candidate embedded platforms. The goal is to select the most efficient and cost-effective platform which enables the implementation of the PIoT.

Modelling

ITI has achieved the following initial results for the use case scenario PRO_UCS5 – Resizing of resources based on current workload.

ITI has been developing new modelling artefacts to accurately represent the dynamic behaviour of the workload of the smart port platform. The smart port platform is composed by: (1) a centralised computing platform responsible of processing the messages coming from the IoT devices and the Terminal Operating System (TOS); and (2) hundreds of these IoT devices and Gateways that are deployed over the port infrastructures, mainly different kind of cranes, trucks and tractors.

The main challenges that have been addressed are twofold. On one side, that the IoT devices, and the corresponding workload they generate, are only active depending on the TOS assignments for a given ship arrival schedule. Modelling this activation intervals requires a second level of scheduling and a set of pseudo-activities that allows art2kitekt tool suite to properly simulate and evaluate the smart port behaviour.

On the other side, the application model that is required to represent the smart port platform is quite large, mainly due to the amount of IoT devices that are involved and the number of modelling artefacts that are needed to activate and deactivate the low-level activities when a ship arrives or departs. To tackle with the introduction of these application models and the new scheduling artefacts a tool for importing large models based on simple spreadsheets have been developed. All the data required to build the model can be easily provided by the UC provider in a custom spreadsheet. The spreadsheet is then processed by this new tool and the resulting models are imported in the art2kitekt tool suite for its evaluation. Figure 8 shows a block diagram of the modelling, simulation and analysis platform the ITI is developing for the smart port use case.

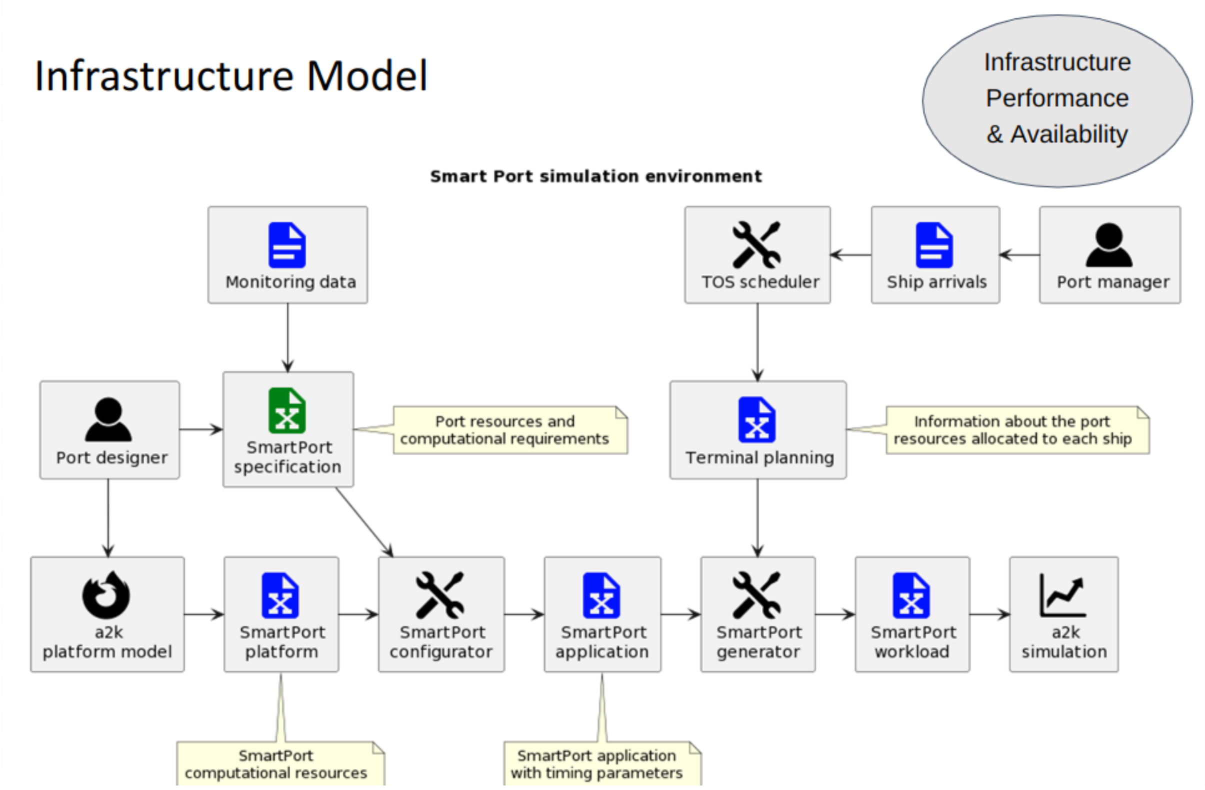

Figure 8 – Modelling of the Smart Port Use Case.

We have also developed and investigated **new algorithms for automatic placement of software services** in a complex cyber-physical system (a2k-optimiser). We represent the system as a graph network of processing elements interconnected by communications links with various properties such as transmission bandwidth, latency, cost, etc. (cf. Figure 9). This model can be generated from the methods described in the previous paragraphs. The software running on the processors is also represented as a graph which models the dependencies of each software service along with the resources it requires, such as its worst-case execution time. We have identified and defined a set of optimisation objectives and constraints to allocate the software services to specific processing nodes. These objectives include, power consumption, end-to-end latency, service spread, and resource usage, amongst others. Constraints include ensuring all the software tasks meet their required deadlines and that the processors are not overloaded. We are using our A2K tool suite to calculate system timing metrics (e.g., task response times) as well as simulation to obtain visual chronograms of multiple tasks sharing a computing resource and thus interfering with each other.

Currently we are evaluating several different multi-objective optimisation solution methods including, for example, genetic and evolutionary algorithms, simulated annealing, and particle swarm algorithms.

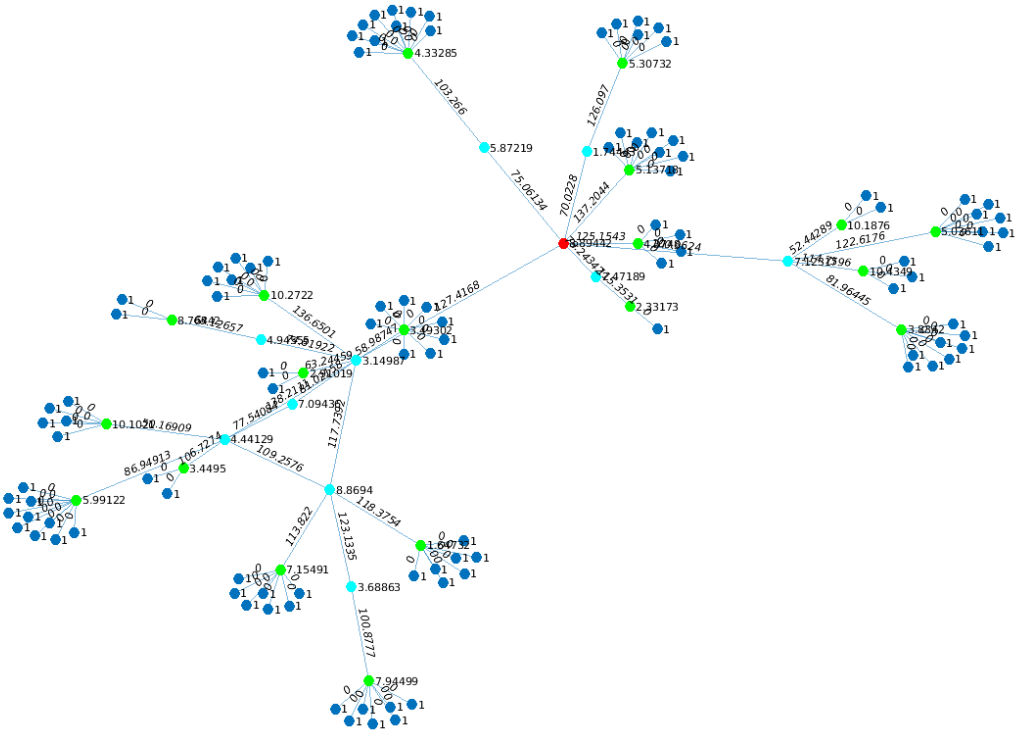

Figure 9 – Graph of Possible Smart Port Network Architecture showing processing nodes (blue: IoT devices, green: gateways, cyan: fog processors, red: cloud servers). The numerical values indicate processing node resource capacities and the latencies of the communications links between nodes.

5.3 TEK Challenge – Design Choice Exploration/Verification for the Prognostics and Health Management (PHM) System – UNIVAQ, UCAN

UNIVAQ

UNIVAQ utilises HEPSYCODE tool for modelling activities related to the Prognostics and Health Management (PHM) system of TEK_UCS_01. We utilise a rendez-vous Process Network Model of Computation (MoC) to model PHM use case, enabling the automatic conversion of high-level models into SystemC functional/timing simulations through a M2T xtext transformation. During the refinement phase, Design Space Exploration (DSE) is employed to find architectural alternative implementations.

To create a complex parallel Artificial Neural Network (ANN) model for the PHM system, we utilise the Sirius-based Graphical User Interface (GUI) of HEPSYCODE, as shown in Figure 10. Through this GUI, we successfully generate executable SystemC code that seamlessly integrates with the HEPSIM2 simulator. Additionally, we conduct preliminary simulations after integrating the ANN code into the model. It is noteworthy that the executable model of TEK_UCS_01 in HEPSYCODE accepts input data conforming to the data models of TEK.

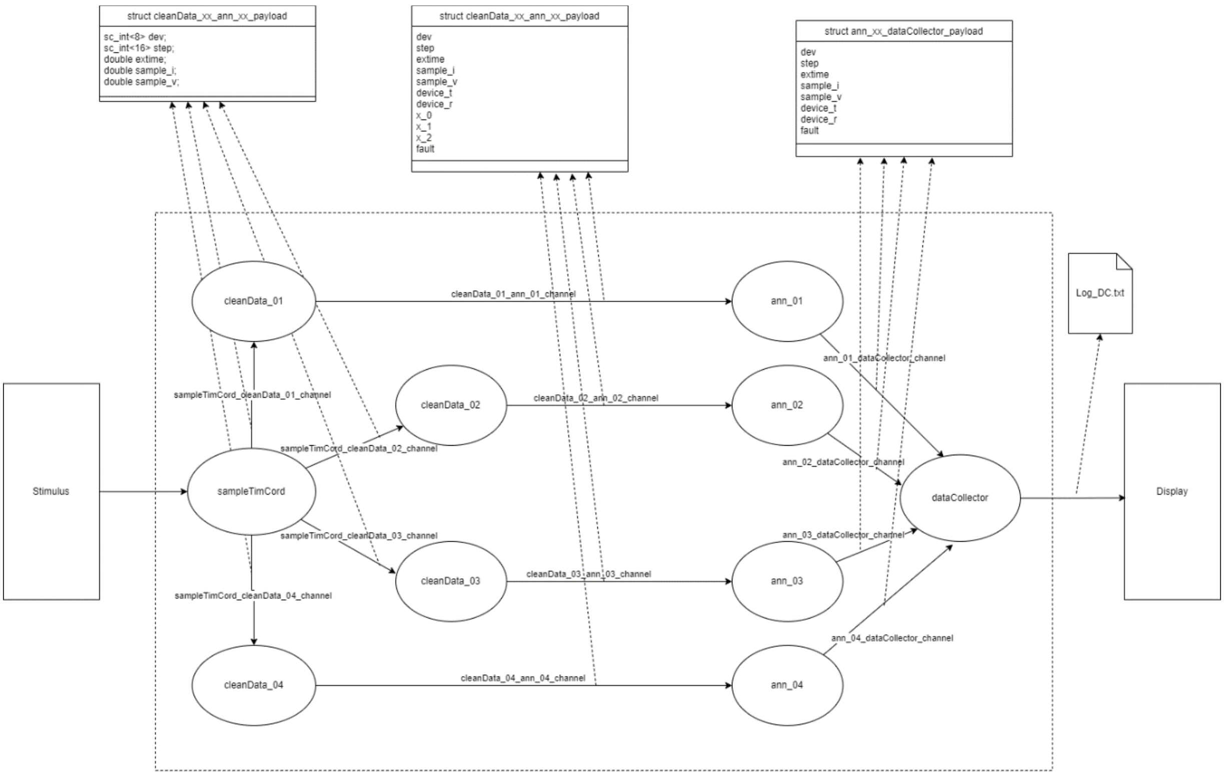

Figure 10 – HEPSYCODE TEK_UCS_01 rendez-vous process network model depicting the execution of 4 ANN processes for the prediction of electrical motor faults.

Furthermore, HEPSYCODE integrates the process mining functionality from JKU to track designer activities during the modelling process. This integration serves to enhance the GUI and improve Sirius features. Additionally, it enables the collection of valuable data for AI/ML modelling analysis and improvements (further work will be performed in the context of WP4 to this respect).

UCAN

The dataset generation methodology developed by UCAN as part of its SoSIM solution is being applied to the TEKNE UCS1. The design of embedded CPS requires models and techniques that encompass subtle interactions between functional and extra-functional concerns. When the functional and temporal interactions between the digital and the physical parts are so important and the development of a physical prototype is not possible or it results too expensive, HW/SW co-design takes relevance and co-simulation becomes essential. Besides the performance simulation, other non-functional concerns apart from timing, such as energy, expands Design-Space Exploration (DSE).

Traditional approaches include cross-compiled simulation. The speed of such a technique is limited by the execution time of the annotation code. Depending on the accuracy required, the annotation code can be larger or smaller. Nevertheless, nowadays, most processors are implemented with HW performance counters. These are special registers within a processor that are used to monitor and measure the performance of the processor and the software running on it. These counters track various metrics, such as instruction counts, cache hits and misses, branch mispredictions, and memory accesses. Performance counters can be used to identify performance bottlenecks, analyse application behaviour, and optimise system performance. In the past, we also found out that host performance counters can be used to predict the performance and energy in a different processor. The analysis was done off-line, once the SW execution in the host was finished.

In this period of the project, we have applied this idea at run-time, i.e. during system simulation. Therefore, the prediction made using the performance counters can be applied dynamically to approximate the execution time of the simulated code. This simplifies the annotation of the code. Whenever the performance prediction and dynamic back-annotation is faster than the execution time of the annotated code, the host performance counters can be a useful source of information for cross-compiled simulation. We have explored the conditions under which Performance Counters Cross-Compiled Simulation (PCCS) results faster than traditional Cross-Compiled Simulation (TCCS).

Our training approach is inspired from the state-of-the-art practice that we adapted to work at runtime. As a result, our process to generate and train the model is sketched in Figure 11.

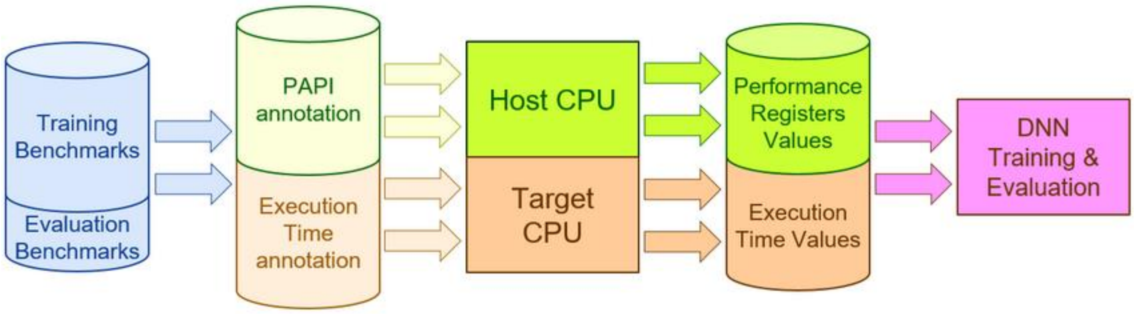

Figure 11 – Dataset generation and DNN training & evaluation.

The time diagram in Figure 12 shows how the trained model is currently being used with CPU and GPU co-processor and being applied for simulation:

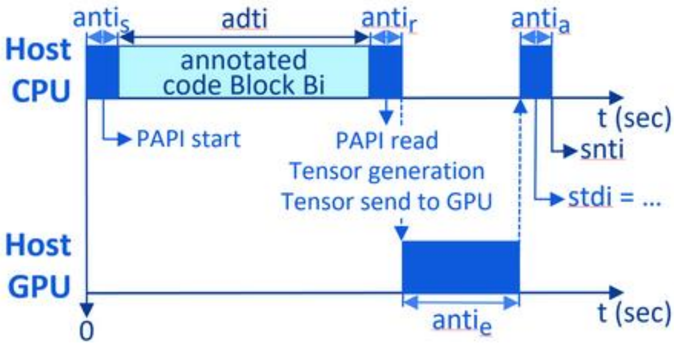

Figure 12 – Distribution of the simulation time in a phase using PAPI.

Regarding the next phase of work in this use case challenge, we have detected a strong dependency of the accuracy with respect to the granularity of the size of the code to be analysed (what we denote as the phase). The NN trained in a first experiment with very good results on large phases is not accurate enough in programs divided in small phases.

A first approach to the dataset generation methodologies have been defined in both the static and the dynamic performance estimation techniques. While the dynamic technique based on PAPIs is almost fixed and only the relation between accuracy and phase size is being analysed, the static technique is going to be completely re-defined in order to make it easier to be applied and to increase its accuracy. The current intention is to link the technique with an ISS simulator.

5.4 ABI Challenge – Modelling and ML for Safety-Critical CPSs in the Automotive Industry – INT

As previously mentioned in this deliverable, INT-XAI is a tool designed to assist ML engineers during the modelling development of an object detector by giving a humanly interpretable reason for the model output (ABI-CH2.c challenge). The object detector model that provides the testing ground is the one underlying the INT-DET solution and is based on a convolutional network (CNN).

Among the latest techniques suitable for this type of model, special attention has been paid to Class Activation Mapping (CAM). The CAM algorithm is a technique that helps to visualize the regions of an image that are relevant for a particular class prediction. It provides insights into what the model is focusing on when making predictions. In the context of an object detector operating in a road driving scenario, such as recognizing pedestrians and vehicles, CAM can be used to explain the decisions made by the model and understand which parts of the image contribute the most to those decisions.

After having trained the CNN-based object detector, the core of INT-DET tool, using a dataset from ABI that contains labelled images of pedestrians, vehicles, and other relevant classes, ongoing work is focusing on integrating CAM into the network architecture: CAM is typically integrated into the final layers of the CNN architecture. These layers often consist of global average pooling (GAP) and fully connected layers. The GAP layer reduces the spatial dimensions of the feature maps to a single vector per channel, while the fully connected layers map the features to class probabilities. The CAM algorithm modifies these final layers to generate class activation maps. The fully connected layers are replaced or augmented with CAM-specific layers. One common approach is to remove the fully connected layers entirely and connect the GAP layer directly to a convolutional layer with a 1x1 kernel. This conversion allows the network to retain spatial information. The output of this 1x1 convolutional layer corresponds to the class activation map. During inference, after passing an input image through the modified network, the class activation maps for each class are obtained. These maps highlight the regions of the input image that contribute the most to the prediction of a specific class. In the road driving scenario, the class activation maps would reveal the areas of the image that the object detector focuses on to identify pedestrians, vehicles, or other objects.

To visualize the CAM results, it is possible to overlay the class activation maps on the original input images. The activation maps are usually up-sampled to match the original image size and then combined with the input image using a colourization scheme (heatmap) where the regions of high activation are represented by warmer colours (e.g., red) and low activation by cooler colours (e.g., blue). By examining these overlaid images, insights into which regions of the image the object detector is attending to when recognizing objects in the road driving scenario can be gained.

By utilizing the CAM algorithm, it is possible to interpret the behaviour of the object detector and verify whether it focuses on the relevant regions of an image for different classes, such as pedestrians, vehicles, or other objects. This visualisation aids in understanding the decision-making process of the model and can help in identifying potential biases, areas of improvement, or areas where the model might be focusing incorrectly.

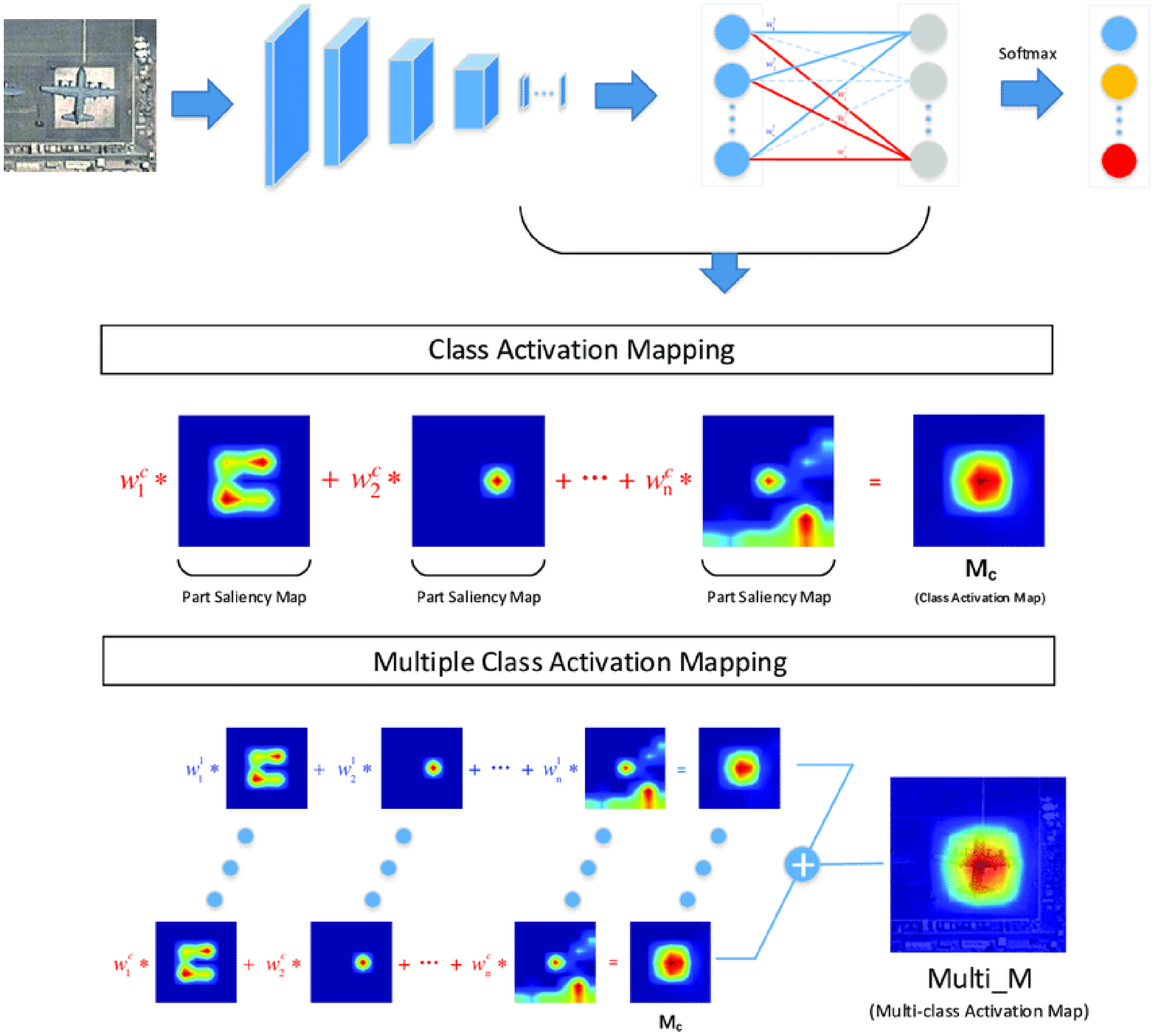

Figure 13 – Comparison between CAM and MultiCAM.

Finally, the application of an extension of the CAM algorithm is currently being studied: the MultiCAM (Multiple Class Activation Mapping) addresses some of CAM limitations and offers additional advantages. While CAM generates class activation maps for a single class at a time, MultiCAM extends this capability to multiple classes simultaneously. This extension enables a more comprehensive understanding of how the model attends to different objects or classes present in an image.

Furthermore, with MultiCAM, the inter-class relationships and interactions within an image can be visualized at once. By examining the activation maps of multiple classes together, insights into how the model discriminates between different objects and the regions where their features overlap or correlate can be gained. This information can be valuable in understanding the decision boundaries and reasoning of the object detector, improving interpretability (cf. Figure 13).

In complex scenes, such as a road driving scenario, MultiCAM can help analyse the relationships between different objects. For example, it is possible to examine how the model attends to both pedestrians and vehicles in an image and identify the regions of potential interaction or proximity between them. This information can be useful for tasks like collision avoidance systems or understanding the behaviour of different objects in a scene. A final advantage is the possibility to mitigate class-specific biases: MultiCAM can aid in identifying and mitigating class-specific biases that may exist in the object detector. By examining the simultaneous activation maps, regions of attention that are disproportionately focused on certain classes or neglecting others can be detected. This insight can guide model improvements and help ensure fair and balanced performance across all classes. These advantages make MultiCAM a valuable tool for analysing and explaining the behaviour of object detectors in complex scenarios like road driving.

6 Conclusion

The work presented in this deliverable D3.4 is in the direct continuation of the WP3 activities that already resulted in the deliverable [AIDOART-D3.2] providing the initial specification of the AIDOaRt Core Tool Set, and the deliverable [AIDOART-D3.3] describing an intermediate version of the AIDOaRt Core Tool Set. The main objective of D3.4 was thus to consolidate the work carried out during the whole duration of WP3 and to refine the description and current status of the AIDOaRt Core Tool Set according to the latest information collected from both use case providers and solution providers. Another goal of D3.4 was also to report on more recent practical applications of solutions from the AIDOaRt Core Tool Set in the context of project's use cases and of their related cyber-physical systems.

At the formal end of WP3, the current situation of the AIDOaRt Core Tool Set in terms of developments and applications is globally positive. Indeed, as visible in this deliverable D3.4 (and also in the previous deliverable [AIDOART-D3.3]) the identified set of solutions allows covering in practice a representative set of related use case challenges and corresponding scenarios. At this stage of the overall project, we have been able to see more and more usages of various (combinations of) solutions from the AIDOaRt Core Tool Set in the context of the use cases. This is satisfying and this effort needs to be continued even after the formal end of WP3.

The latest work reported in this deliverable D3.4 is important as it serves as input to the integration effort currently carried out in the context of WP5 (with other related inputs coming from WP2 and W4). Moreover, D3.4 reports on some recent collaborations between solutions providers and use case providers, with the goal to foster future potential similar collaborations before the end of the project and even afterwards. As far as WP3 is concerned, D3.4 is the last formal deliverable. However, the work on the different solutions from the AIDOaRt Core Tool Set and the various collaborations with the use case providers on their various challenges and scenarios will of course continue until the complete termination of the project. Concretely, from now on, the consolidation and practical applications of these solutions will be performed in the context of both WP4 (for the AI-related aspects of these solutions) and WP5 (for the integration of these solutions in the overall AIDOaRt framework and in the different use cases). As a consequence, the corresponding updates and new results will be directly integrated into the future WP4 and WP5 deliverable whenever relevant.

7 References

[AIDOART-D1.1] AIDOaRt consortium: Use cases requirements specification, 2021. (Access restricted to the AIDOaRt project's consortium).

[AIDOART-D1.2] AIDOaRt consortium: Architecture Specification – Initial Version, 2021. (Access restricted to the AIDOaRt project's consortium).

[AIDOART-D1.3] AIDOaRt consortium: Use cases scenarios and evaluation criteria definition, 2021. (Access restricted to the AIDOaRt project's consortium).

[AIDOART-D1.4] AIDOaRt consortium: Architecture Specification – Final Version, 2022. (Access restricted to the AIDOaRt project's consortium).

[AIDOART-D2.1] AIDOaRt consortium: AIDOaRt Core Infrastructure and Framework – Initial Version, 2021. (Available online at: https://www.aidoart.eu/download/18.7d39d59b181260bc66b12343/1654680192979/D2.1.pdf).

[AIDOART-D2.2] AIDOaRt consortium: AIDOaRt Core Infrastructure and Framework – Intermediate Version, 2022. (Available online at: https://www.aidoart.eu/download/18.3022ca0a184b9251f0a105bb/1670242565559/AIDOaRt%20D2.2%20Data%20Collection%20and%20Representation%20-%20Interim%20Version.pdf).

[AIDOART-D3.1] AIDOaRt consortium: Report on Foundations of MDE and AIOPS for DevOps, 2021. (Available online at: https://www.aidoart.eu/download/18.1b4dc71217eaa69c2fa5499d/1644265570830/D3.1.%20Report%20on%20Foundations%20of%20MDE%20and%20AIOPS.pdf).

[AIDOART-D3.2] AIDOaRt consortium: AIDOaRt Core Infrastructure and Framework – Initial Version, 2022. (Available online at: https://www.aidoart.eu/download/18.7d39d59b181260bc66b1233f/1654680145439/D3.2.pdf).

[AIDOART-D3.3] AIDOaRt consortium: AIDOaRt Core Infrastructure and Framework – Intermediate Version, 2022. (Available online at: https://www.aidoart.eu/download/18.3022ca0a184b9251f0a105bd/1670242714911/AIDOaRt_D3-3%20_AIDOaRt-Core-Infrastructure-and-Framework-Intermediate-Version.pdf).

[AIDOART-D4.1] AIDOaRt consortium: AIDOaRt AI-Augmented toolkit – Initial Version, 2022 (Available online at: https://www.aidoart.eu/download/18.3022ca0a184b9251f0a105c0/1670243232895/D4.1%20AIDOaRt%20AI-Augmented%20toolkit%20-%20%20Initial%20Version.pdf).

[AIDOART-D4.2] AIDOaRt consortium: AIDOaRt AI-Augmented toolkit - Intermediate Version, 2023 (Available online soon).

[AIDOART-D4.3] AIDOaRt consortium: AIDOaRt AI-Augmented toolkit - Final Version, 2023-2024 (Not available yet).

[AIDOART-D4.4] AIDOaRt consortium: AIDOaRt AI-Augmented toolkit - Guidelines, 2023-2024 (Not available yet).

[AIDOART-D5.1] AIDOaRt consortium: AIDOaRt Integration Approach, 2022 (Available online at: https://www.aidoart.eu/download/18.7d39d59b181260bc66b12340/1654680145599/D5.1.pdf)

[AIDOART-D5.2] AIDOaRt consortium: AIDOaRt Integrated framework - initial version, 2022 (Not available yet).

[AIDOART-D5.3] AIDOaRt consortium: AIDOaRt Integrated framework - intermediate version, 2023 (Access restricted to the AIDOaRt project's consortium).

[AIDOART-D5.4] AIDOaRt consortium: AIDOaRt Integrated framework - final version, 2024 (Not available yet).

[AIDOART-D5.5] AIDOaRt consortium: Use cases requirements and scenarios evaluation report, 2022 (Available online at: https://www.aidoart.eu/download/18.7d39d59b181260bc66b12341/1654680145791/D5.5.pdf).

[AIDOART-D5.6] AIDOaRt consortium: Use cases development report - 1, 2022 (Access restricted to the AIDOaRt project's consortium).

[AIDOART-D5.7] AIDOaRt consortium: Use cases evaluation report - 1, 2023 (Not available yet).

[AIDOART-D5.8] AIDOaRt consortium: Use cases development report - 2, 2024 (Not available yet).

[AIDOART-D5.9] AIDOaRt consortium: Use cases evaluation report - 2, 2024 (Not available yet).