AIDOaRt: AI-augmented Automation for DevOps, a Model-based Framework for Continuous Development in Cyber-Physical Systems

Hugo Bruneliere*

IMT Atlantique, LS2N (UMR CNRS 6004), Nantes, France

Vittoriano Muttillo, Romina Eramo

University of L'Aquila, L'Aquila, Italy

Luca Berardinelli

Johannes Kepler University, Linz, Austria

Abel Gómez

Universitat Oberta de Catalunya, Barcelona, Spain

Alessandra Bagnato, Andrey Sadovykh

SOFTEAM, Paris, France

Antonio Cicchetti

Mälardalen University, Västerås, Sweden

Abstract

The advent of complex Cyber-Physical Systems (CPSs) creates the need for more efficient engineering processes. Recently, DevOps promoted the idea of considering a closer continuous integration between system development (including its design) and operational deployment. Despite their use being still currently limited, Artificial Intelligence (AI) techniques are suitable candidates for improving such system engineering activities (cf. AIOps). In this context, AIDOaRT is a large European collaborative project that aims at providing AI-augmented automation capabilities to better support the modelling, coding, testing, monitoring, and continuous development of CPSs. The project proposes to combine Model Driven Engineering principles and techniques with AI-enhanced methods and tools for engineering more trustable CPSs. The resulting framework will 1) enable the dynamic observation and analysis of system data collected at both runtime and design time and 2) provide dedicated AI-augmented solutions that will then be validated in concrete industrial cases. This paper describes the main research objectives and underlying paradigms of the AIDOaRt project. It also introduces the conceptual architecture and proposed approach of the AIDOaRt overall solution. Finally, it reports on the actual project practices and discusses the current results and future plans.

Keywords: Cyber-Physical Systems, Continuous Development, System Engineering, Software Engineering, Model Driven Engineering, Artificial Intelligence, DevOps, AIOps.

*Corresponding author

Email addresses: hugo.bruneliere@imt-atlantique.fr (Hugo Bruneliere), {vittoriano.muttillo,romina.eramo}@univaq.it (Vittoriano Muttillo, Romina Eramo), luca.berardinelli@jku.at (Luca Berardinelli), agomezlla@uoc.edu (Abel Gómez), {alessandra.bagnato,andrey.sadovykh}@softeam.fr (Alessandra Bagnato, Andrey Sadovykh), antonio.cicchetti@mdu.se (Antonio Cicchetti)

January 17, 2023

1. Introduction

Modern complex Cyber-Physical Systems (CPSs) [1, 2] in the areas of Industry 4.0, healthcare, autonomously driving cars or smart grids are highly interconnected systems where software enables increasingly advanced capabilities [3, 4]. Such CPSs imply new challenges in their engineering process, from the design, development or analysis phases to the actual deployment, usage, future maintenance, and evolution. In this context, Model Driven Engineering (MDE) [5, 6] is a relevant Software Engineering paradigm to raise the abstraction level and improve the ability to handle complexity. The use of models, as first-class abstractions of systems and environments, is a fundamental element for technologies in current and future CPS engineering platforms [1].

With the advent of DevOps [7, 8], CPS engineering would also benefit from continuous development approaches proposing a smooth continuum from design to runtime (and vice versa). In parallel, IT leaders envision the productivity boost of tomorrow to be brought by the application of Artificial Intelligence (AI) principles and techniques [9]. Thanks to AI for IT operations (AIOps) [10], the standard DevOps pipeline can be rethought by allowing continuous monitoring, alerting and remediation in a more securely and reliably. However, at this stage, we have not been able to find any generic and reusable AI-augmented approach intending to support full continuous CPS engineering.

The AIDOaRt project [11, 12] aims at providing a novel MDE framework based on AI-augmentation to efficiently support the continuous software and system engineering of CPSs. Complementary to the support for existing systems, AIDOaRt also aims to improve the continuous development of new modern CPSs. AIDOaRt is a 3-year H2020-ECSEL European collaborative project that started on April 1, 2021, involving 32 organizations grouped in national clusters from 7 different countries. Its novelty mainly lies in the combination of MDE, AI/ML and DevOps [13]. Thanks to that, DevOps teams may use the AIDOaRt model-based framework to analyze event streams (for real-time and historical data), together with design information available as models, to extract meaningful insights on the CPS and take proactive decisions. Such information can then be used for continuous improvement, faster deployment, better collaboration, reduced downtime, etc. We expect an industrial uptake of AIDOaRt results via developing real complex and large-scale CPSs.

This paper is a significantly extended version of our already published DSD 2021 conference paper [11]. The previous content has been revised by adding much more details on the AIDOaRt project's consortium, scientific and technical background, conceptual architecture, supporting approach, etc. We also added a completely new section detailing the actual project practices, including the general project structure (in terms of work packages), the global strategy and its updated timeline, and the day-to-day process we follow in the project. Finally, we largely reworked and extended the last section of the paper (before the conclusion) with the description of the project's current status, main ongoing activities and possible future impact after nearly one year of execution.

The rest of the paper is structured as follows. Section 2 presents the AIDOaRt consortium. Section 3 introduces the project's main objectives, research and technical background, and conceptual architecture. Then, Section 4 describes the AIDOaRt overall approach and its different core components. Section 5 provides an overview of the industrial use cases for deployment and experimentation; Then, Section 6 explains the project practices currently carried out in AIDOaRt. Section 7 reports on the current status of the work, the ongoing activities at the end of Year 1 of the project, and the plan and future impact for the coming couple of years. Finally, Section 8 concludes the paper.

2. Project Consortium

When setting up the AIDOaRt consortium shown in Fig. 1, a great effort has been made to ensure a well-balanced alliance of European partnerships among academia, SMEs, large industry, and research organisations. It brings together a balanced consortium of 32 European partners (7 Large Companies, 12 SMEs, and 13 Academic and R&D Organizations). These partners are geographically covering 7 different European countries (Austria - AT, Czech Republic - CZ, Finland - FI, France - FR, Italy - IT, Spain - ES, and Sweden - SE).

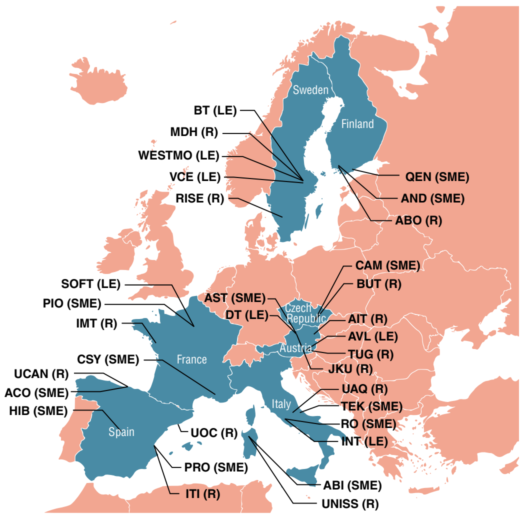

Figure 1: Map of the AIDOaRt Partners

MDU, as the project coordinator, guarantees an adequate level of management based on their extensive experience in coordinating such large research projects. They are supported by several partners who have already worked together in the past with great success (notably in the

MegaM@Rt2 project [14]) along with complementary partners newly integrated into AIDOaRt. This team of experienced partners is directly connected to the different communities of researchers, experts and stakeholders in the advanced fields of AI, MDE and DevOps as the core scientific and technical topics of the project. The AIDOaRt consortium offers an adequate combination of use-case providers, technology/service providers, and R&D providers.

2.1. Use Case Providers

Ten industrial partners will play the role of case study providers and end-users. The technological domains and case study owners are summarized in Table 1. The proposed use cases are presented in more detail in Section 5.

2.2. Technology/Service/Solution Providers

Nine industrial partners will play the role of technology and service providers, while thirteen partners will drive the research activities. Their names and contributions to the project are summarized in Table 2 and Table 3.

3. Objectives and Concepts

This section presents the main AIDOaRt objectives (cf. Section 3.1) as well as the research and technical areas of the conceptual framework to realize the envisioned goals (cf. Section 3.2 and Section 3.3, respectively).

3.1. AIDOaRt Objectives

The information recovered from system execution, either large historical data or real time traces, is of very high value to assess the validity and properties of system design. However, it requires dedicated methods and tools to extract useful knowledge as well as to analyze and reuse it in a productive way. To this end, AIDOaRt aims at integrating AI innovations to ensure that systems are correctly designed and to contribute to our trust in their behavior.

As show in Table 4, the project aims at creating a framework incorporating methods and tools for continuous software/system engineering [18] and validation, by leveraging the advantages of AI techniques. We expect to observe a significant improvement in productivity, quality, and predictability of CPSs.

3.2. AIDOaRt Background

As introduced before, AIDOaRt aims at combining MDE principles and techniques with AI technologies to improve the continuous development of CPSs, i.e., going from DevOps to AIOps.

Model Driven Engineering (MDE). MDE allows raising the level of abstraction and thus improving the ability to engineer and handle complex systems such as CPSs [1]. The use of models as purposeful abstractions of systems and environments is also increasing within the industry (e.g., digital twining [19]). While first-generation MDE tools mainly focused on generating code from high-level models, they now also address model-based testing, verification, measurement, tool/language interoperability or software evolution, among many other software engineering challenges. In AIDOaRt, MDE will contribute by 1) providing better abstraction principles and techniques (e.g., for the handled data), 2) facilitating the automation of DT engineering activities, and 3) supporting technology integration among all the covered design and development activities.

Artificial Intelligence and Machine Learning (AI/ML). The dissemination of AI principles and techniques in a regulated industry can rapidly enable systems to decide and act in a more and more automated manner, sometimes even without direct human control [20]. Consequently, this demands a responsible approach to ensure the safe and beneficial use of AI technologies. This approach has to consider both the implications of (co)decision-making by machines and related ethical issues (cf. the definition of AI legal status [21]). Within AIDOaRt, we aim at 1) ensuring that systems are designed responsibly and 2) contributing to our trust in their behavior. This notably requires supporting both **explainability** [22] (i.e., being able to explain and justify decisions), and **accountability** [23] (i.e., justifying, based on explanations, decisions/actions taken by stakeholders involved in the CPS engineering process).

DevOps. AIDOaRt also aims to contribute to the adoption of DevOps [24, 8, 25], a software engineering paradigm focused on software delivery by enabling continuous feedback and quick response to changes, and using automated delivery pipelines resulting in reduced cycle time. In particular, tools and methods focus on administration and automation processes. In AIDOaRt, we are proposing an overall model-based approach combining existing techniques to cover more efficiently the different DevOps phases (cf. also AIOps).

Artificial Intelligence for IT Operations (AIOps). AIOps [9, 26] characterise solutions where DevOps challenges are addressed with the help of AI/ML techniques. A fundamental challenge notably resides in integrating AIOps [10] in enhancing the DevOps pipeline, e.g., through continuous alert and insights from data used for continuous deployment and operations management. AIOps not only uses data science and computational techniques to automate common and routine operational tasks but also digests metrics and uses inference models to extract actionable insights from data. We believe AI is a relevant complementary mean, along with MDE, to achieve AIDOaRt goals.

Data. From an MDE perspective, everything is a model, including all kinds of data. In AIDOaRt, data will be collected both at runtime and design time, and relevant traceability information between these two levels will be computed and maintained. Once available, all this data needs to be handled using a combination of MDE and AI techniques. To this end, all relevant data will be treated

Table 1: Case studies from AIDOaRt Partners

Domain	Partner	Co.	**Description**
Automotive	Abinsula SRL - ABI	IT	Safety Critical Systems in the Automotive Domain using Disruption Technology
Automotive	AVL List GmbH - AVL	AT	AI-supported Digital Twin Synthesis Supporting Secure Vehicle Development and Testing for Novel Propulsion Systems
Railway	Bombardier Transportation - BT	SE	DevOps for Railway Propulsion System Design
Traffic	CAMEA, spol. s.r.o. - CAMEA	CZ	AI for Traffic Monitoring Systems
Railway	CLEARSY SAS - CSY	FR	Machine learning in interactive proving
Catering	HI Iberia Ingeniería y Proyectos S.L. - HIB	ES	AI DevOps in the Restaurants Business
Maritime	Prodevelop SL - PRO	ES	Smart Port Platform Monitoring
Electronic	Tekne SRL - TEK	IT	Agile process and Electric/Electronic Architecture of a Vehicle for Professional Applications
ICT Services	Volvo Construction Equipment AB - VCE	SE	Data Modeling to Support Product Development Cost and Efficiency
TLC	Westermo Network Technologies AB - WESTMO	SE	Automated Continuous Decision Making in Testing of Robust and Industrial-grade Network Equipment

Table 2: Technology/Service Providers from AIDOaRt Industrial Partners

Partner	Co.	**Description**
ACORDE Technologies - ACO	ES	Solutions over continuum computing exploiting AI/ML for monitoring
Anders Innovations Oy - AND	FI	DevOps pipeline tool in software security, testing, delivery, and monitoring by using machine learning
Automated Software Testing - AST	AT	Knowledge improvement in AI, process improvement in agile-DevOps, and methodology improvement in MDE development and MBT
Prevision.io - PIO	FR	AutoML platform in order to easily automatize modeling and prediction
Qentinel Oy - QEN	FI	Cloud-based infrastructure for testing using the Qentinel Pace platform
Ro Technology - ROTECH	IT	Development processes on complex systems through continuous development and continuous integration
Dynatrace Austria - DT	AT	Keptn and Dynatrace product, with AIOps approach and DevOps methods
Intecs Solutions - INT	IT	Create new business opportunities in various markets (e.g., automotive, railway, TLC)
Softeam - SOFT	FR	Bring the acquired knowledge to integrate the sustainable use of AI in CPS

Table 3: Solution providers from AIDOaRt R&D Partners

Partner	Co.	**Description**
Åbo Akademi - ABO	FI	Applying AI and ML to novel software engineering methods and tool
Universitat Oberta de Catalunya - UOC	ES	Model-based technologies (e.g., TemporalEMF, AsyncAPI or WAPIml)
Institut Mines-Telecom - IMTA	FR	Model-based approaches and tools, notably EMF Views and ATL
Maelardalens Hoegskola - MDH	SE	Cross-fertilisation between the various research groups at MDH
Technische Universitaet Graz - TUG	AT	Combine automata learning with model-based test case generation
Universidad de Cantabria - UCAN	ES	S3D extension (i.e., Single-Source System Design framework)
Università degli Studi dell'Aquila - UNIVAQ	IT	Engineering processes in the design of CPSs [15, 16] and the combination of MDE techniques with AI/ML [17]
Università degli Studi di Sassari - UNISS	IT	Solutions for consistency checking of requirements, verification of DNNs and automatic test generation
Johannes Kepler University Linz - JKU	AT	Research on MDE, DevOps, and AI/ML by exploiting complementary areas
Brno University of Technology - BUT	CZ	New AI components into two areas of activity - cutting-edge research and education
Austrian Institute of Technology - AIT	AT	Model-based V&V techniques and the application of AI to bridge the gap between development and operations through new methods and tools
Instituto Tecnológico de Informática - ITI	ES	Improvement of the art2kitekt framework in combination with other consortium partners
Research Institutes of Sweden - RISE	SE	AI/ML-based testing and analysis solutions for various industrial use cases from different domains

Table 4: AIDOaRt specific objectives

Obj.	**Description**
O1	**Holistic integration of system data.** To support the collection, representation and tracing of different heterogeneous data sources covering both runtime and design time data *(Observe)*, their combined analysis based on previous observations *(Analyze)*, and the automation of DevOps pipeline tasks from the previous analysis *(Automate)*.
O2	**Model-based framework for the continuous development of CPS.** To develop a scalable framework for efficiently handling numerous, heterogeneous, and large models of the system under development. We will rely on results from the previous MegaM@Rt2 [14] project to be suitable for large distributed cross-functional teams,
O3	**AI-augmented toolkit.** To support an enhanced DevOps toolchain integrating AI techniques (notably Machine Learning) in multiple aspects of the system engineering process. This is to be realized via a combination of AIOps and MDE, and by extending the AIDOaRt core model-based framework.
O4	**Demonstrators validation -** To develop specific demonstrators showing the applicability and validity of the AIDOaRt model-based framework and AI-augmented toolkit through 8 complementary industrial case studies (cf. Section 5).
O5	**European excellence in Continuous System Engineering and AIOps.** The AIDOaRt project incorporates an innovative research agenda and work plan that will contribute to maintaining and further advancing the AI-augmented and DevOps world-class research that already exists in Europe (cf. Section 2).

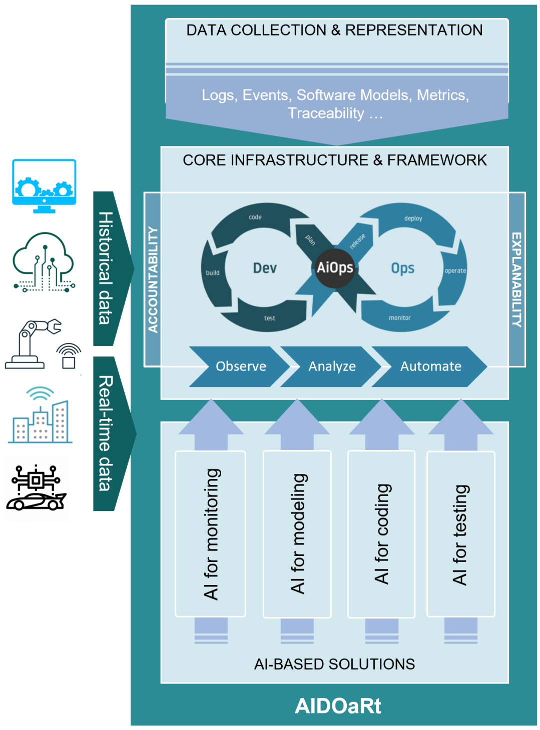

Figure 2: AIDOaRt Conceptual Architecture

as models, enabling share and (re)use in different contexts and by different technical solutions of the framework (including AI ones) [27]. Moreover, this will allow supporting interoperability between different data formats.

3.3. AIDOaRt Conceptual Architecture

Fig. 2 provides a conceptual architecture of the AIDOaRt global solution and emphasizes its key principles and concepts as introduced before.

The overall AIDOaRt infrastructure consumes different kinds of data, including notably runtime data (e.g., IT monitoring, log events, etc.) and design data produced during the software development process (e.g., software models, design documentation, traceability information, source code, etc.). All the collected data will be processed, translated into an internal model-based representation, and then collected in a shared repository (*Data Collection & Model-based Representation* component. The *Model-based Core Infrastructure* component is intended to support the standard DevOps practices by efficiently combining software development and IT operations.

Moreover, AIDOaRt aims to enhance DevOps toolchains (cf. existing DevOps tools [28]) by employing AI and ML techniques in multiple aspects of the system development process (e.g., requirements, monitoring, modeling, coding, and testing). This capability will be made available as an *AI-augmented Toolkit*. In an AIOps-enabled context, the toolkit should support 1) the monitoring of runtime data (such as logs, events and metrics [15]) and software data and their traceability (namely *Observe*), 2) the analysis of both historical and real-time data (namely *Analyze*), and 3) the automation of development and operation activities (namely *Automate*). Accountability [23] and explainability [22] capabilities should be managed as cross-cutting concerns throughout the DevOps engineering process and toolchain. These capabilities will consume available design-time and runtime data which, according to MDE principles, should be made available to stakeholders as design-time and runtime models, respectively.

Finally, AIDOaRt aims to extend existing techniques and introduce novel solutions. Notably, the state of the art of requirements engineering, monitoring, and testing, already includes mechanisms supporting/leveraging data analysis [29, 30, 31]. Moreover, search-based techniques have been investigated to automate MDE-related activities such as language engineering, model transformation, and model versioning [32]. Nonetheless, especially when dealing with mission-critical systems, the automated generation of artifacts raises verification and validation issues, e.g., for certification purposes [33].

4. The AIDOaRt Approach

The main innovation in AIDOaRt involves combining DevOps with advanced MDE and AI principles and techniques. The AIDOaRt conceptual architecture from Fig. 2 is detailed in an activity-like diagram in Fig. 3.

Operational steps (blank rounded boxes) and input/output data/artifacts (blank squared boxes) are interconnected in a continuous flow of steps and corresponding input/output data/artifacts. A hierarchy of grey round boxes further aggregates steps and data/artifacts flowing in the envisioned AIDOaRt components (data collection and representation, core framework and infrastructure, AI-augmented toolkit) and typical phases of a DevOps cycle, from plan to monitor, applied to the given CPS.

During system development, several data/artifacts are produced at runtime and design time. At this level, we do not distinguish the nature of these data/artifacts: they can be models, text, databases, etc. We consider all these as the data set that the AIDOaRt *Data collection and representation* component acquires, elaborates, and transforms into *Data models* that conform to a common *Data metamodel*. Such Data models are then given as inputs to the *Core framework and infrastructure* component that provides the core model-based capabilities and services to support the *Observe*, *Analyze* and *Automate* general operations. For example, it includes core model management services like storage [34], editing/handling/querying [35], transformation [36], views [37] and eventually code generation [38]. The *Core framework and infrastructure* specifies and provides common interfaces/APIs to access these services and some concrete tools implementing these APIs.

Finally, the AIOps operations provided by the *AI-augmented toolkit* component, powered by the AIDOaRt-core framework and infrastructure features, support some typical engineering activities targeted in AIDOaRt,

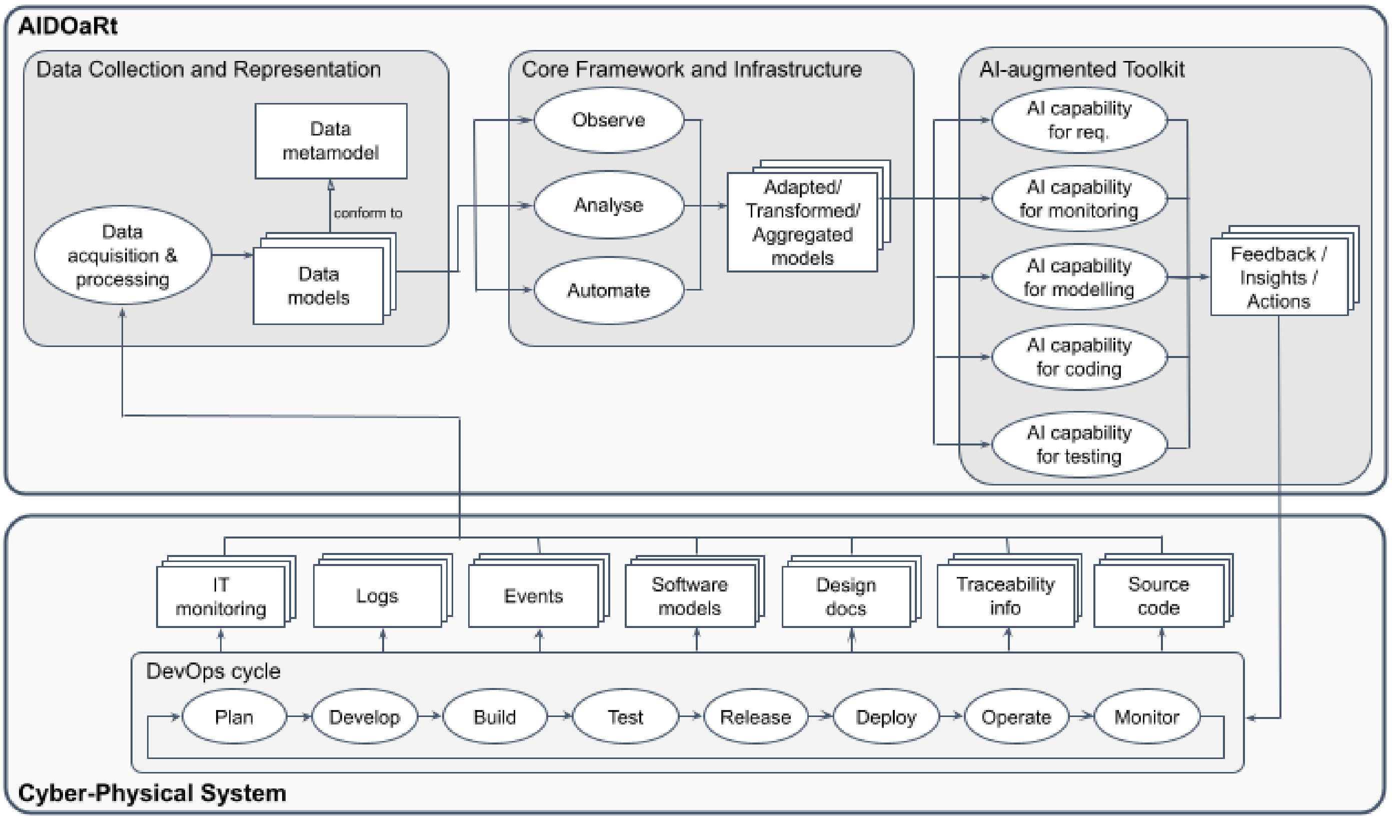

Figure 3: AIDOaRt Approach

namely requirements, monitoring, modeling, coding, and testing ones. The output of these operations consists of feedback, insights, or actions to be exploited in specific phases of the DevOps cycle.

To realize these contributions, we notably plan to base our work on previous initiatives from different past and present European projects [14, 39]. The objective is to 1) reuse (parts of) the existing approaches or prototypes whenever relevant and 2) both complement and integrate them with new solutions designed and developed in AIDOaRt.

4.1. AIDOaRt Data Collection and Representation

The AIDOaRt *Data Collection and Representation* tooling is being designed and developed to support the management of the collected data that represents the initial input required by the AIDOaRt framework. The combination of real-time and historical data, along with data coming from system design, will allow the AIDOaRt framework to predict certain CPS properties and help make relevant decisions during the CPS engineering process. A successful framework for AIOps solutions must able to collect data from an entire multi-vendor and multi-domain environment, including network and storage solutions, containers, and cloud infrastructure. As already introduced earlier, data are collected both at runtime (with IT monitoring, events logging, etc.) and at design time (from software models, design documentation, traceability information, source code, etc.). Thus, we are developing methods and supporting tools to acquire, correlate and provide access to a broad range of historical and real-time data types.

Historical data is collected information about past events or artifacts. It includes documents, artifacts, (software) models, runtime data, and traceability information. Real-time data is information that is delivered immediately after collections, without delay in the timeliness of the information provided. It mainly refers to runtime data like logs, events, and metrics collected on the running system under continuous development. AIDOaRt aims to gather as much information as possible about system design, architecture, behavior, or events in previous time periods. based on that, it can provide adequate insights and background for making decisions that will ultimately affect the course of the development. To this end, relevant data sets are created and handled by the AIDOaRt framework. In particular, collected data will be cleaned, analyzed, and translated to an internal data model (i.e., the internal representation) by relying on a combination of MDE and AI techniques. One or more dedicated metamodels or megamodel/repositories are used as an internal representation. All the components and services of the core framework and the toolkit must refer to the AIDOaRt internal representation.

To realize this in practice, AIDOaRt aims at reusing existing data management solutions (i.e., data platforms that permit real-time processing of streaming IT data like

Hadoop 2.0, Elastic Stack, or some Apache technologies) that will be integrated within the AIDOaRt methods and tools. In this respect, the value of the AIOps solutions lies in their ability to ingest and correlate data across the environment. Runtime and design time data collected by the AIDOaRt framework have to preserve relevant traceability information among them, notably to provide proper and reusable feedback from runtime to design time.

4.2. AIDOaRt Core Infrastructure and Framework

The AIDOaRt *Core Infrastructure and Framework* is being designed and developed to support the core capabilities to be provided in the project context. The AIDOaRt infrastructure provides the base platform of AIDOaRt, which consists of computational and storage services to support the targeted engineering activities (i.e., requirements, monitoring, modeling, coding, and testing). The infrastructure exposes the AIDOaRt internal data models as obtained in the previous step, providing the model-based representation for every kind of data needed in the framework.

As a result, the end-users can invoke an extensible set of AIDOaRt tools that, built on top infrastructural services, support the realization of the engineering activities and specific analysis, e.g., for explainability or accountability, according to the global AIDOaRt architecture. Going more into detail, the AIDOaRt Core Infrastructure and Framework consist of a common, modular and extensible environment relying on various kinds of software components/tools. It first provides generic and customizable support for the AIDOaRt AIOps process, i.e., the CPS continuous development process to be augmented on-demand with complementary AI-based extensions (cf. the AIDOaRt AI-augmented Toolkit as described in Section 4.3). In addition, it offers communication interfaces / APIs with the AIDOaRt Data Collection and Representation, i.e., capabilities for importing (and then consuming) historical and real-time data models resulting from the CPSs under development (cf. Section 4.1). It also offers extension interfaces/APIs intended to the AIDOaRt AI-augmented Toolkit (cf. Section 4.3), i.e., the support for plugging and integrating AI-augmented capabilities related to several CPS development tasks (e.g., requirements, monitoring, modeling, coding, and testing).

Moreover, the AIDOaRt Core Framework includes methods and tools for specific services, like accountability or explainability, that incorporate ethical principles and addressing societal concerns. *Accountability* can be defined as 1) a quality or state of an acting subject responsible for giving an account, i.e., capable of answering a given act, 2) a quality or state of a passive object that is capable of being accounted for (by an active subject), 3) a quality or state of an activity (performed by given subjects and acting on given objects), which is continuously observable and reportable. In AIDOaRt, we consider accountability as a quality or state of engineering activities following DevOps and MDE principles. Therefore, continuously evolving MDE artifacts represent the keystone engineering data through which the system under study (i.e., the objects) and the involved stakeholders (i.e., the subjects) are made accountable. *Explainability* can be considered the prerequisite/enabling concept for accountability. It is a term coined in the context of Explainable Artificial Intelligence (XAI) to describe the ability to explain the decision process of AI-controlled entities, usually regarded as black-box entities. The goal of enabling explainability in AI/ML is to ensure that algorithmic decisions, as well as any data driving those decisions, can be explained to end-users and other stakeholders in non-technical terms. In the AIDOaRt context, explainability is considered as the capability, within a given decision/engineering process, to generate explanations to stakeholders involved in the decision process of the concerned CPS, e.g., architects, modelers, programmers, testers, IT experts, as well as to recommend actions to be taken to improve the system life cycle. Following MDE principles, such an explanation can be expressed as a model [20].

To conclude, the AIDOaRt Core Infrastructure and Framework has to provide all the means used in the AI-augmented tools to perform various continuous system engineering tasks described in the next section.

4.3. AIDOaRt AI-augmented Toolkit

To extend the AIDOaRt *Core Infrastructure and Framework*, complementary AI-augmented capabilities must provided and reused accordingly. To this end, the AIDOaRt *AI-augmented Toolkit* is being designed and developed in the project context. It offers additional capabilities related to different CPS development tasks[1]. It will also provide some use case-specific extensions developed in the project to be later used as demonstrators of the practical capabilities of the AIDOaRt solution (cf. Section 5).

The *AI-augmented Toolkit* aims to collect, keep, and use CPS data coming from a various sources including runtime metrics, events, topology, log files as well as design models and measures (cf. Section 4.1). It should notably support the typical AIOps activities. This starts with the *Ingestion and Handling* of data, events, and metrics from many different sources via solutions for monitoring and solutions for data management (for example). Then, this covers *Engagement and Analysis* by employing AI/ML for pattern discovery and prediction, anomaly detection, root cause determination, and data correlation and aggregation. Specific feedback and insight will be obtained to support one or more development phases (requirements, monitoring, modeling, coding, and testing). Finally, the *Automation* of development process operations is supported by leveraging prescriptive advice (e.g., pattern-based prediction)

[1]Some important tasks such as deployment are not explicitly mentioned here. However, they will be facilitated by the general automation capabilities we plan to provide.

to take automated action (i.e., remediation, proactive operation).

AI for requirements. AI/ML methods will support requirement elicitation, recommendation, and consistency checking at the early design stage. To help the elicitation, we plan to consider historical requirement documents and exploit AI/ML for supporting activities at later stages of the development process. AI techniques will also recommend (types of) requirements considering the concerned application domain and its related standards and guidelines. Another important aspect that will be investigated is requirements consistency checking by relying on Automated Reasoning (AR) techniques. Finally, we will also develop natural language processing (NLP)-based tools to support the writing and management of requirements.

AI for monitoring. We will focus on AI-based tools and techniques to deal with run-time properties verification, system fault detection and forecasting for predictive maintenance. These tasks will be accomplished by means of AI/ML algorithms that, considering the result of trace analysis and related historical data, will allow detecting both performance and system errors.

AI for modeling. AI-based tools and techniques will support decision-making, strategy planning, and recommendations during the modeling activities. In particular, there will be a developed AI-based assistant that guides the engineer in the design of complex mixed-critical systems. Moreover, AI/ML techniques will be considered to support the (semi-)automated creation and/or update of large and complex model views involving heterogeneous (design time and runtime) models [40]. Several AI-based techniques will also be used to support model refinement and verification, notably ML and AR for design consistency checking, automated properties/specifications verification, anomaly detection, and root cause determination in temporal behavior.

AI for coding. From an MDE perspective, code can be considered an executable textual model. Nevertheless, AI-based tools targeting the coding phase will also be developed to help users with no or very superficial programming experience create programs. In this case, AI/ML techniques will be mainly employed to learn code generation patterns from examples.

AI for testing. Finally, we will develop AI-based tools for testing at different levels. For example, AI/ML techniques will be proposed to gather, examine, and observe data models to generate reliable unit tests. Pattern recognition techniques will also be developed to identify high quality test suites, e.g., set of tests with more failure probabilities.

4.4. AIDOaRt Integration

To apply and evaluate in practice the AIDOaRt *Core Infrastructure and Framework* in combination with the AIDOaRt *AI-augmented Toolkit*, an integration approach is being specified and developed. The proposed integration is notably based on the use of standards for sharing the various underlying models/data and efficiently combining the different tools resulting from AIDOaRt technical components (described in Sections 4.1 4.2 and 4.3). The use of MDE artifacts and methodologies does not limit the specific integration architecture to be used. On the contrary, using model-based representations of the data provided a common understanding no matter how the data is exposed and consumed, e.g., data lakes, APIs, etc. For instance, existing APIs can be refactored (e.g., extended, reduced, combined, or split) by discovering and linking the models behind each individual API. On the one hand, thanks to this flexibility, the AIDOaRt integration approach aims at creating a global API exposing a unified data model by merging the data models of the initial APIs, as defined in the *Core framework* and exploited in the *AI-augmented toolkit*. On the other hand, AIDOaRt will be able to adapt the global API to the specific use cases based on data and processing capabilities required for performing their AI-related operations and workflows.

A global integration architecture has been defined based on the constraints and requirements of the use case providers and solution providers. To tame the risk of a complex technical integration due to an expected technological heterogeneity, the integration explicitly considers customizable engineering processes relying on common MDE principles and practices (e.g., metamodels, models, and transformations). For this purpose, technological interoperability is already enabled through the tight collaboration between academic partners, technology providers, and industrial partners in the project consortium.

5. Industrial Use Cases

AIDOaRt brings together prominent technology providers and research organizations with state-of-the-art methods and tools to be validated in highly relevant European industry case studies. The end users from the space, naval, railway, smart grid, smart warehouse, and telecom industry domains will drive the project by providing real-world requirements and case studies and validating and endorsing the AIDOaRt research and technical results.

The AIDOaRt Core Infrastructure and Framework will be applied in different industrial use cases, as presented in Table 1. The variety of use cases will validate the different AI-augmented extensions of the AIDOaRt Toolkit, i.e., requirements management, monitoring, modeling, coding, and testing.

As a practical illustration, one of the industrial use cases is provided by Volvo Construction Equipment (VCE UC). VCE is undergoing a transformation in its product development context: they are evolving from traditional complex mechanical systems with embedded software towards more advanced software and technology-intensive units. Both software and mechanical units are the building blocks of variants of VCE product lines, e.g., Volvo articulated

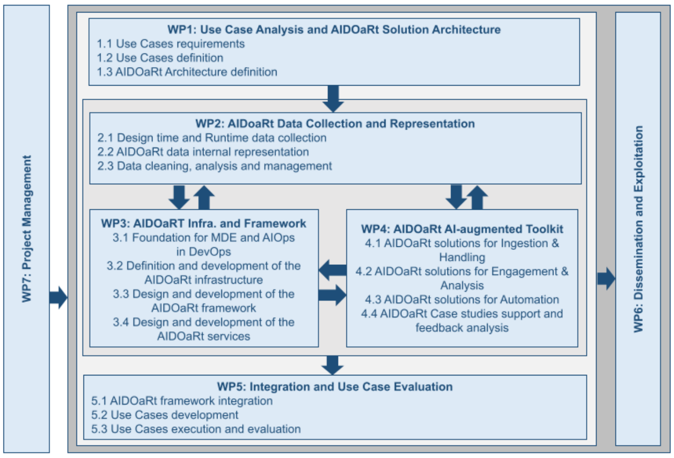

Figure 4: Work packages structure

haulers [41]. Therefore, VCE is introducing MDE techniques and practices in its system engineering process. In particular, the VCE use case envisions the adoption of suitably interwoven design models based on standardized languages (e.g., SysML [42], AutomationML [43]) and simulation models based on simulation standards (e.g., FMI [44]). To support the seamless evolution of current VCE engineering processes and maximize the acceptance of new MDE-based practices, we are currently working on supporting AI for modeling solutions based on i) model transformations to exchange engineering knowledge among existing non-MDE artifacts (Excel spreadsheets, Visio drawings) and prescriptive engineering models (in SysML and AutomationML), and ii) extending AIDOaRt and VCE tool sets with modeling process mining and recommendation capabilities.

6. Project Practices

This section presents the actual AIDOaRt practices in terms of work organization (cf. Section 6.1), global strategy and timeline (cf. Section 6.2), as well as day-to-day process and activities (cf. Section 6.3).

6.1. Overall Work Package Structure

The AIDOaRt project is structured into seven complementary work packages (WPs), each encapsulating a coherent body of work (cf. Fig. 4 and Table 5 for more details). The first five WPs cover the research and technical activities to be carried out during the project in collaboration between the different partners. The last two WPs cover the transverse activities running through the whole project, which concern the management, dissemination, and exploitation of the various work and results obtained in the first five WPs.

6.2. Global Strategy and Timeline

Research activities are inherently risky: it is hardly possible to predict the results and plan every detail of the technologies or artifacts to be developed. Thus, we adopted a planning approach that enables pivoting and adjusting whenever necessary. Milestone-driven planning with regular planning waves appeared to be an appropriate solution for a 3-year project of that scale. Indeed, milestones define the important decision points when certain results are available, and the near future becomes clearer. In AIDOaRt, these decision points are the Architecture Design / Foundations and several consecutive phases of Use Case Evaluation (Fig. 5). Reaching these milestones requires specific R&D activities such as analyzing the state-of-the-art concerning research and technological areas or the release of specific technical components.

Based on the AIDOaRt organization in terms of WPs, Fig. 5 illustrates the sequence of the main activities carried out during the project's execution. To realize this in practice, the project follows an iteration path (cf. Section 6.3) for three main technology-related activities. First, the **Data framework engineering** methods and techniques for the data collection and management are designed, developed, integrated, and validated/refined based on the use case requirements and scenario definition (i.e., runtime and design data involved in the use cases). Then, **Core infrastructure and framework engineering** capabilities are designed, developed, integrated, and refined/validated based on the use case requirements specification (first version) and the progress realized within the use cases under development (next versions). Finally, the **AI-augmented toolkit engineering** features are designed, developed, and integrated with the rest of the AIDOaRt solution. The toolkit is then refined/validated based on the use cases under development.

Each activity is organized in three distinct phases implementing associated DevOps work items. In the **Initial** iteration, the focus is put on the prototyping and incremental research on top of the baseline technologies. Another important aspect is matching the existing technologies, both from the state-of-the-art and proposed by AIDOaRt solution providers, with the concrete requirements elicited by the use cases. Technologies start to be designed, implemented, integrated, and then validated based on the use case requirements and related scenarios. The intermediate version of the AIDOaRt solutions is built In the **Intermediate** iteration, , and more corresponding research results are expected. In particular, the design of the solutions is improved based on the feedback received from the initial iteration. In parallel, the use cases are practically developed for evaluation, and corresponding feedback is provided. In the **Final** iteration, technologies are consolidated based on the received feedback, and the use cases development is finalized. This third iteration ends with the provisioning of the final integration and evaluation of the AIDOaRt solutions in the context of the project's use cases and related scenarios.

6.3. Day-to-day Continuous Process

We adopted a flexible timeline that accommodates the unpredictable nature of research activities. Therefore, on a

Table 5: AIDOaRt technical work-packages organization.

WP	Description
WP1	**Use Cases Analysis and AIDOaRt Solution Architecture** sets up the context and specification for the technological development in the project by analyzing the use case requirements. The analysis is also meant to ensure the applicability, reusability, and interoperability of the results across different applications and domains.
WP2	**AIDOaRt Data Collection and Representation** supports the management of the collected data, both real-time and historical, as initial input for the AIDOaRt solution. The data from the system engineering process (e.g., design models) is also considered to predict some outcomes and make decisions accordingly.
WP3	**AIDOaRt Infrastructure and Framework** provides the general infrastructure, framework, and core services of the AIDOaRt solution. It consists of computational, networking, and storage capabilities as well as common APIs and interfaces to support DevOps engineering activities in an ethical and socially-aware way.
WP4	**AIDOaRt AI-augmented Toolkit** develops the AI-augmented tool set according to the needs of various kinds of CPSs (as defined by the use cases). As a combination of AIOps and DevOps in a MDE setting, it targets the application of AI for requirements, monitoring, modeling, coding, testing, and deployment.
WP5	**Integration and Use Case Evaluation** coordinates the integration of research and technical developments from WP2, WP3 and WP4. This WP provides specific industrial demonstrators via use cases from the various industry domains involved in the project, and provides a form of validation and KPIs measurement.

daily basis, the project is carried out via short-term micro-tasks inspired by agile practices and following a continuous integration / continuous delivery (CI/CD) approach inspired by DevOps. These tasks can evolve and be adjusted at any point in time by considering the need for the overall development and release process. Project partners are mostly distinguished as technology providers and use case providers. This way, it is possible to effectively coordinate the activities, match requirements and available solutions, and also better clarify the actual target for each specific task.

The iteration path is depicted in Fig. 6. The **Technology plan** work item 1 includes the requirements definition and use cases scenarios specification, as well as the development plan and strategy according to the AIDOaRt global architecture. Specific attention is given to the characteristics of the data collected by use case providers and to the available services/components offered by solution providers. The **Technology development** work item 2 concerns the development of the various technical elements necessary to compose the AIDOaRt solution. The **Technology integration** work item 3 concerns the effective combinations of the different technical components to obtain the final AIDOaRt solution. The **UC development** work item 4 concerns the practical application of the AIDOaRt solution in the various use cases. In other words, research partners and solution providers contribute to transferring the developed technologies to use case providers. The **UC execution and validation** work item 5 includes the execution and validation of the results from work items 1 to 3 based on the work performed in the context of work item 4. The **Evaluation & feedback** work item 6 includes the evaluation of the result of all previous work items (and particularly work item 5) to provide relevant feedback to be used in the next iteration of the whole process.

7. Current Status, Plans and Future Impact

This section describes the results already obtained after nearly one year of AIDOaRt and the currently ongoing work (cf. Section 6.1). It also provides insights into the general plan and longer-term objectives in expected industrial impact (cf. Section 7.2).

7.1. Results of Year 1 and Ongoing Work

At the time of writing, the project has just ended its first year of execution (thus still having two years to go). In what follows, we describe the various results produced in the already running WPs and the ongoing work.

WP1. Year 1 of the project has been dedicated to 1) the definition of the use cases requirements, corresponding scenarios, and criteria for their future evaluation and 2) the specification of the architecture of the overall AIDOaRt solution. To realize this, we heavily relied on a model-based information collection, management and integration process that facilitated documented production [45]. This resulted in three complementary deliverables released by the end of 2021, whose access is restricted to the consortium members, namely D1.1, D1.2, and D1.3. They will be followed by a second and final version of the architecture of the overall AIDOaRt solution to be released as D1.4 during Year 2 of the project. The work in WP1 is performed in close collaboration with all the partners also strongly involved in the technical WPs, notably the technology providers. This commitment is fundamental as a

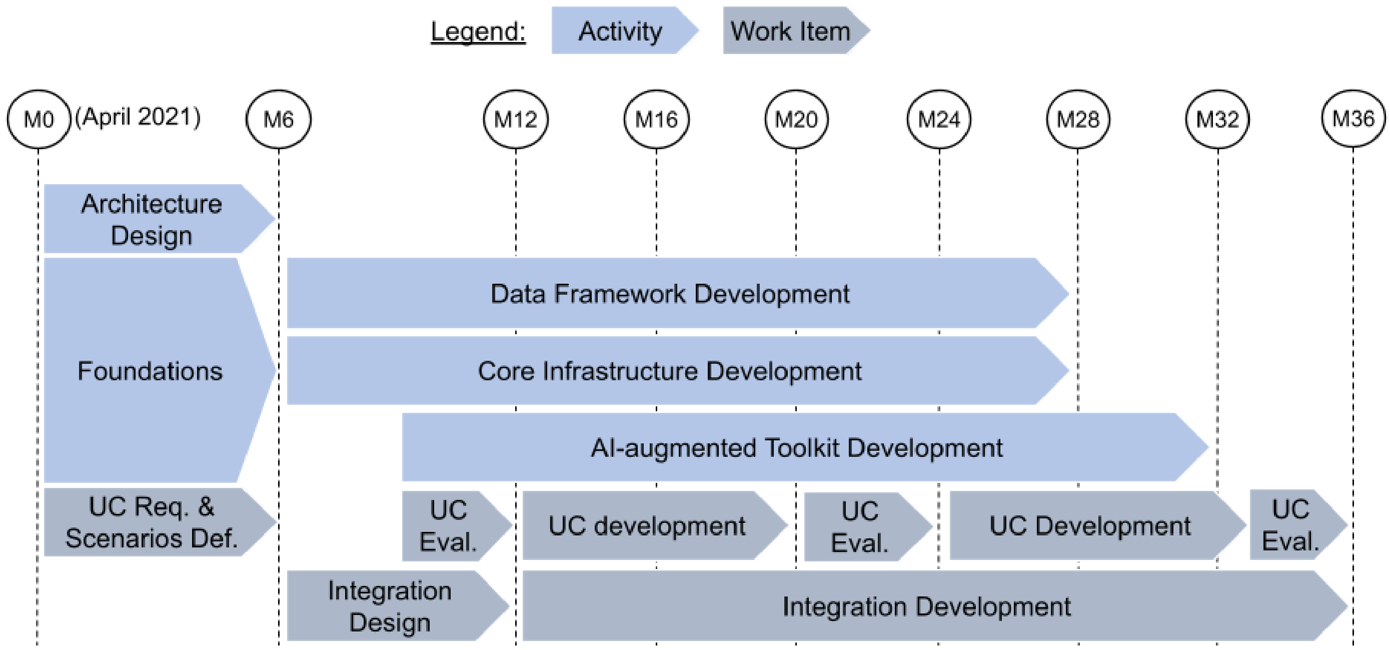

Figure 5: AIDOaRt Timeline

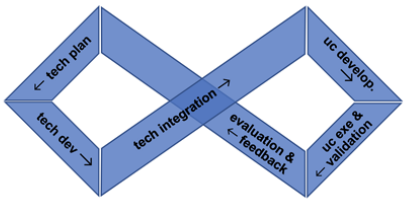

Figure 6: The DevOps iteration path

significant part of the information gathered in the context of WP1 is also used to support the work on the WP2-WP3-WP4 and WP5 deliverables.

WP2. The work focused on identifying the data-related needs that could be expressed by both technology providers and use case providers. This has been done directly with WP1 concerning the use cases requirements definition and the first version of the AIDOaRt architecture specification. 31 tools have been declared as potentially supporting the AIDOaRt Data Collection and Representation framework: 11 tools targeting *Data Collection*, 9 tools for *Data Management* and 11 tools *Data Representation*. The initial version of the AIDOaRt Data Collection and Representation tool set has been released as deliverable D2.1 at the end of M12. Two other versions of this tool set will follow, respectively as deliverable D2.2 in Year 2 and D2.3 in Year 3. A current research effort focuses on consolidating the data models from the different use cases and provided tools. The objective is to produce a corresponding AIDOaRt megamodel that links these data models together and allows them to be accessed in a centralized and unified way [46].

WP3. The work started with a deep analysis of the state-of-the-art concerning the main scientific and technological areas of interest in AIDOaRt, i.e., MDE, AI/ML, and DevOps. Concretely, this study has been conducted as a Systematic Mapping Study (SMS) following standard guidelines [47, 48, 49]. The main goal was to classify existing contributions from the scientific literature that combine MDE, AI/ML, or DevOps concerns to support the continuous development of complex systems. An initial set of 1152 potentially relevant papers has been collected from bibliographic sources. Among these, 133 primary studies have been selected and then analyzed to characterize and classify their contributions in engineering processes, involved resources and engineered products [50]. This work proposes a baseline to gain insights on these areas, assess the novelty of the outcomes from the different WPs, and shape challenges and future research directions. This resulted in deliverable D3.1 [13] and an updated version of this SMS is currently under finalization to be submitted for publication. In parallel to this, the work also started on specifying the initial version of the AIDOaRt Core Infrastructure and Framework. 28 tools have been declared as potentially supporting this Core Infrastructure and Framework (a single tool can possibly target different capabilities): 11 tools are targeting *Infrastructure*, 24 tools *Data and Modeling* and 5 tools *Generic Services*. The initial version of the AIDOaRt Core tool set was released as deliverable D3.2 at the end of M12. Two other versions of this tool set will follow, as deliverable D3.3 in Year 2 and deliverable D3.4 in Year 3.

WP4. The work focused on identifying the available tools to be possibly considered and then integrated to achieve the objectives in terms of AI/ML support for DevOps processes. Thus, the work already started on specifying the initial version of the AIDOaRt AI-Augmented toolkit. Currently, several tools have been declared as potentially supporting this AI-Augmented toolkit (a single tool can possibly target different capabilities): 8 tools can contribute to *Ingestion and Handling*, 17 tools to *Engagement and analysis*, and 12 tools to *Automation*. Among these, 4 tools target the support for *Monitoring*, 7 tools the support for *Requirements*, 13 tools the support for *Modelling*, 6 tools the support for *Coding*, and 16 tools the support for *Testing*. The initial version of the AIDOaRt AI-Augmented tool set has been be released as deliverable D4.1 at the end of M16. Two other versions of this tool set will follow: deliverable D4.2 in Year 2 and D4.3 in Year 3.

WP5. The work started only from M6 of the project, as strongly relying on previous efforts performed in parallel in the context of WP1 (use case definition and architecture specification) and WP3 (state-of-the-art). At this stage, the activities focused on preparing the future integration of the different tools, components, models, etc. designed and developed in the context of WP2-WP3-WP4. Notably, a set of integration patterns have been identified and described as different ways of possibly combining the project's scientific and technological results in order to meet the use case requirements. Concretely, this results in deliverable D5.1. AIDOaRt Integration Approach, which was released at the end of M12. It will then be followed by three versions of the actual AIDOaRt integrated framework (or tool sets) to be progressively released in Year 2 and Year 3 as deliverables D5.2, D5.3, and D5.4.

7.2. Plan and Industrial Impact

On the one hand, AIDOaRt targets the architecture, design, and integration of complex and heterogeneous CPSs. The underlying challenges gravitate around the management of critical, autonomous, cooperating, and evolving systems. They are also concerned with managing the complexity, diversity, and multiple constraints of such CPSs. On the other hand, the project targets the support for better connectivity and interoperability both inside and outside such CPSs. Notably, data interoperability or safe, secure, trustable, and reliable connectivity between the involved components are key aspects we consider. The transverse character of the project will allow its deployment and impact on currently important industrial application domains such as Transport and Smart Mobility, Digital Industry (a.k.a. Industry 4.0) or Digital Life (e.g., Smart Cities, Smart Societies).

AIDOaRt plan to reach its goals by capitalizing on the expertise gained by research organizations and industrial partners in prior projects, notably the MegaM@Rt2 project, which proposed tools and methodologies for design-runtime interaction and feedback loop in continuous system development [14, 39]. As a result, the AIDOaRt approach and solutions are expected to be applied to complex and heterogeneous use cases and to impact the automotive, transportation, technology, and trade sectors (as covered by the partners' business and case studies, cf. Section 5). According to the PWC auditing and consulting [51], these particular sectors are expected to be among the most influenced ones (first together with health, fourth, fifth, and sixth, respectively) in the short (0-3y), medium (3-7y), and long term (>7y). AIDOaRt is expected to extensively contribute to two of the most challenging sectors (i.e., automotive and transportation) by focusing on potential productivity gains obtained by higher automation of the system and software engineering processes. Interestingly, this is also a fundamental goal of both AIOps (AI applied to DevOps) and MDE. The project results will not be limited to the above-mentioned domains and are expected to contribute to a cross-domain ecosystem proposing integrated AI-augmented solutions for the generic support of system and software engineering.

The AIDOaRt achievements will be measured using quantitative and rigorous parameters based on both metrics defined in the literature and data collected by partners during the project development (notably in the context of WP1 and WP5, cf. Section 6). Additional evaluations will be considered concerning the relative improvements that will be achieved when compared with already existing/applied technologies and practices within the involved companies. To this end, we plan to provide a reliable measurement approach that identifies and treats accordingly the normal variations that might occur in design, development, and operation activities. We hope that such a global evaluation can be easily understood both by project partners and potential users of the AIDOaRt solutions from outside of the consortium.

Finally, special attention will be paid to scenarios where improvements provided by the AIDOaRt solutions are difficult or complex to quantify. This will also be the case when significant subjective elements affect the industrial domain, the offered capabilities, or the experience of the persons carrying out the evaluation. Other criteria may be equally important since they represent key aspects, e.g., ease of use, maturity level, etc., that can substantially impact the adoption of AIDOaRt results. This kind of improvement is expected to be quantified by satisfaction measures typically calculated using questionnaires and interviews regarding certain capabilities and experiences associated with using the AIDOaRt approach and solutions.

8. Conclusion

AIDOaRt is a 3-year H2020-ECSEL European project involving 32 organizations grouped in national clusters from 7 different European countries. Its major objective is to strengthen European excellence by providing a model-based framework to more efficiently support the continuous software and system engineering of CPSs via the AI-augmentation of key activities such as modeling, coding, testing, and monitoring. The framework will offer a holistic integration of different kinds of system data, i.e., design time and runtime data collected from both the system engineering process and the system's actual execution. The proposed approach and supporting solution will be deployed and evaluated on a set of industrial use cases from several different application domains as demonstrators of the genericity and validity of the developed technologies.

At the end of Year 1, we already delivered the first concrete outcomes in the form of detailed use case definitions and related practical scenarios, a study of the state-of-the-art in the concerning scientific and technical areas, a first version of the architecture of the AIDOaRt overall approach/solution as well as an initial specification of the three tool sets aiming at supporting this architecture. All these results will be consolidated and extended in the coming two years with the final objective of evaluating them in real industrial CPS engineering processes. Other assets, e.g., scientific papers reporting on related research contributions, are also planned to be published complementary to a final and stable version of the architecture of the AIDOaRt overall approach/solution and next iterations of the three AIDOaRt tool sets.

Acknowledgments

Special thanks to the AIDOaRt consortium members. The AIDOaRt 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, and Spain.

References

[1] H. Thompson, M. Reimann, D. Ramos-Hernandez, Platforms4CPS, Key Outcomes and Recommendations, Steinbeis-Edition, 2018.

[2] L. Pomante, V. Muttillo, B. Křena, T. Vojnar, F. Veljković, P. Magnin, M. Matschnig, B. Fischer, J. Martinez, T. Gruber, The aquas ecsel project aggregated quality assurance for systems: Co-engineering inside and across the product life cycle, Microprocessors and Microsystems 69 (2019) 54–67.

[3] C. Sau, C. Rinaldi, L. Pomante, F. Palumbo, G. Valente, T. Fanni, M. Martinez, F. van der Linden, T. Basten, M. Geilen, G. Peeren, J. Kadlec, P. Jääskeläinen, L. Bulej, F. Barranco, J. Saarinen, T. Säntti, M. K. Zedda, V. Sanchez, S. T. Nikkhah, D. Goswami, G. Amat, L. Maršík, M. van Helvoort, L. Medina, Z. Al-Ars, A. de Beer, Design and management of image processing pipelines within cps: Acquired experience towards the end of the fitoptivis ecsel project, Microprocessors and Microsystems 87 (2021) 104350.

[4] J. Agirre, L. Etxeberria, R. Barbosa, S. Basagiannis, G. Giantamidis, T. Bauer, E. Ferrari, M. Labayen Esnaola, V. Orani, J. Öberg, D. Pereira, J. Proença, R. Schlick, A. Smrčka, W. Tiberti, S. Tonetta, M. Bozzano, A. Yazici, B. Sangchoolie, The valu3s ecsel project: Verification and validation of automated systems safety and security, Microprocessors and Microsystems 87 (2021) 104349.

[5] D. C. Schmidt, Guest editor's introduction: Model-driven engineering, Computer 39 (2) (2006) 25–31.

[6] M. Brambilla, J. Cabot, M. Wimmer, Model-Driven Software Engineering in Practice, Second Edition, Morgan & Claypool Publishers, 2017.

[7] C. Ebert, G. Gallardo, J. Hernantes, N. Serrano, Devops, IEEE Software 33 (3) (2016) 94–100.

[8] R. Jabbari, N. bin Ali, K. Petersen, B. Tanveer, What is devops? a systematic mapping study on definitions and practices, in: XP2016, XP '16 Workshops, ACM, New York, NY, USA, 2016, pp. 1–11.

[9] S. G. Charley Rich, Pankaj Prasad, Market guide for aiops platforms, id g00378587, Tech. rep., Gartner Research (2019).

[10] Y. Dang, Q. Lin, P. Huang, Aiops: Real-world challenges and research innovations, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2019, pp. 4–5.

[11] R. Eramo, V. Muttillo, L. Berardinelli, H. Bruneliere, A. Gomez, A. Bagnato, A. Sadovykh, A. Cicchetti, AIDOaRt: AI-augmented Automation for DevOps, a Model-based Framework for Continuous Development in Cyber-Physical Systems, in: DSD 2021, Palermo, Italy, 2021, pp. 303–310.

[12] AIDOaRt ECSEL-JU - AI-augmented automation supporting modeling, coding, testing, monitoring, and continuous development in Cyber-Physical Systems, `https://www.aidoart.eu/` (accessed: 28.07.2022).

[13] AIDOaRt D3.1 - Report on Foundation of MDE and AIOPS for DevOps, `https://hdl.handle.net/20.500.12004/1/P/AIDOART/D3.1` (accessed: 28.07.2022).

[14] W. Afzal, H. Bruneliere, D. Di Ruscio, A. Sadovykh, S. Mazzini, E. Cariou, D. Truscan, J. Cabot, A. Gómez, J. Gorroñogoitia, et al., The MegaM@ Rt2 ECSEL project: MegaModelling at Runtime–Scalable model-based framework for continuous development and runtime validation of complex systems, Microprocessors and Microsystems 61 (2018) 86–95.

[15] G. Valente, T. Fanni, C. Sau, T. Di Mascio, L. Pomante, F. Palumbo, A composable monitoring system for heterogeneous embedded platforms, ACM Transactions on Embedded Computing Systems (in press).

[16] T. Di Mascio, F. Caruso, L. Tarantino, G. Valente, Monica vision: An approach, a model and the interactive tools for cyber-physical systems designers, in: CHItaly 2021, 2021, pp. 1–5.

[17] L. Pomante, V. Muttillo, M. Santic, P. Serri, Systemc-based electronic system-level design space exploration environment for dedicated heterogeneous multi-processor systems, Microprocessors and Microsystems 72 (2020) 102898.

[18] B. Fitzgerald, K.-J. Stol, Continuous software engineering: A roadmap and agenda, Journal of Systems and Software 123 (2017) 176–189.

[19] F. Bordeleau, B. Combemale, R. Eramo, M. van den Brand, M. Wimmer, Towards model-driven digital twin engineering: Current opportunities and future challenges, in: ICSMM 2020, Springer, 2020, pp. 43–54.

[20] M. Blumreiter, J. Greenyer, F. J. Chiyah Garcia, V. Klös, M. Schwammberger, C. Sommer, A. Vogelsang, A. Wortmann, Towards self-explainable cyber-physical systems, in: MODELS 2019 Companion, 2019, pp. 543–548.

[21] Gartner Predicts the Future of AI Technologies, `https://www.gartner.com/smarterwithgartner/gartner-predicts-the-future-of-ai-technologies/` (accessed: 28.07.2022).

[22] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.

[23] V. Dignum, Responsible autonomy, CoRR abs/1706.02513. `arXiv:1706.02513`.

[24] L. Leite, C. Rocha, F. Kon, D. Milojicic, P. Meirelles, A survey of devops concepts and challenges, ACM Comput. Surv. 52 (6).

[25] B. Combemale, M. Wimmer, Towards a model-based devops for cyber-physical systems, in: Second international workshop on software engineering aspects of continuous development and new paradigms of software production and deployment, Vol. 12055, Springer, 2019, pp. 84–94.

[26] C. R. Manjunath Bhat, Augment decision making in devops using ai techniques, id g00383246, Tech. rep., Gartner Research (2019).

[27] T. B. Pedersen, Managing big multidimensional data: A journey from acquisition to prescriptive analytics, in: J. Bernardino, C. Quix, J. Filipe (Eds.), DATA 2017, SciTePress, 2017, p. 5.

[28] N. Ceresani, The periodic table of devops tools v.2 is here, https://blog.xebialabs.com/2016/06/14/periodic-table-devops-tools-v-2/, last accessed on 30.06.2021 (Jun. 2016).

[29] W. Maalej, M. Nayebi, T. Johann, G. Ruhe, Toward data-driven requirements engineering, IEEE Software 33 (1) (2016) 48–54.

[30] S. Wolny, A. Mazak, M. Wimmer, C. Huemer, Model-driven runtime state identification, in: H. C. Mayr, S. Rinderle-Ma, S. Strecker (Eds.), EMISA 2019, Vol. P-304 of LNI, Gesellschaft für Informatik e.V., 2019, pp. 29–44.

[31] L. C. Briand, Novel applications of machine learning in software testing, in: QSIC '08, 2008, pp. 3–10.

[32] I. Boussaïd, P. Siarry, M. Ahmed-Nacer, A survey on search-based model-driven engineering, Automated Software Engineering 24 (2017) 233–294.

[33] F. Leofante, N. Narodytska, L. Pulina, A. Tacchella, Automated verification of neural networks: Advances, challenges and perspectives, in press (2018).
URL http://arxiv.org/abs/1805.09938v1

[34] J. Di Rocco, D. Di Ruscio, L. Iovino, A. Pierantonio, Collaborative repositories in model-driven engineering [software technology], IEEE Software 32 (3) (2015) 28–34.

[35] Z. Ujhelyi, G. Bergmann, Á. Hegedüs, Á. Horváth, B. Izsó, I. Ráth, Z. Szatmári, D. Varró, Emf-incquery: An integrated development environment for live model queries, Science of Computer Programming 98 (2015) 80–99.

[36] N. Kahani, M. Bagherzadeh, J. R. Cordy, J. Dingel, D. Varró, Survey and classification of model transformation tools, Software & Systems Modeling 18 (4) (2019) 2361–2397.

[37] H. Bruneliere, E. Burger, J. Cabot, M. Wimmer, A feature-based survey of model view approaches, Software & Systems Modeling 18 (3) (2019) 1931–1952.

[38] G. Sebastián, J. A. Gallud, R. Tesoriero, Code generation using model driven architecture: A systematic mapping study, Journal of Computer Languages 56 (2020) 100935.

[39] MegaMart2 ECSEL-JU - An scalable model-based framework for continuous development and runtime validation of complex systems, https://megamart2-ecsel.eu/ (accessed: 28.07.2022).

[40] H. Bruneliere, F. M. de Kerchove, G. Daniel, S. Madani, D. Kolovos, J. Cabot, Scalable model views over heterogeneous modeling technologies and resources, Software and Systems Modeling 19 (4) (2020) 827–851.

[41] Articulated Haulers - Volvo Construction Equipment, https://www.volvoce.com/global/en/our-offer/articulated-haulers/ (accessed: 28.07.2022).

[42] Systems Modeling Language, https://sysml.org/ (accessed: 28.07.2022).

[43] AutomationML, https://www.automationml.org/ (accessed: 28.07.2022).

[44] Functional Mock-up Interface (FMI), https://fmi-standard.org/ (accessed: 28.07.2022).

[45] A. Sadovykh, D. Truscan, H. Bruneliere, Applying model-based requirements engineering in three large european collaborative projects: An experience report, in: RE 2021, IEEE, 2021, pp. 367–377.

[46] J. Bézivin, F. Jouault, P. Valduriez, On the need for megamodels, in: Workshop on Best Practices for Model-Driven Software Development, OOPSLA/GPCE 2004, Citeseer, 2004, pp. 1–9.

[47] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: EASE 2008, 2008, pp. 1–10.

[48] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and Software Technology 64 (2015) 1–18.

[49] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: EASE 2014, 2014, pp. 1–10.

[50] M. Schleipen, R. Drath, Three-view-concept for modeling process or manufacturing plants with AutomationML, ETFA 2009 (2009) 1–4.

[51] PWC - What's the real value of AI for your business and how can you capitalise?, https://www.pwc.com/gx/en/issues/analytics/assets/pwc-ai-analysis-sizing-the-prize-report.pdf (accessed: 28.07.2022).