

Towards Automating Model-Based Systems Engineering in Industry - An Experience Report

Johan Cederbladh, Luca Berardinelli, Hugo Bruneliere, Antonio Cicchetti, Mohammadhadi Dehghani, Claudio Di Sipio, James Pontes Miranda, Abbas Rahimi, Riccardo Rubei, Jagadish Suryadevara

▶ **To cite this version:**

Johan Cederbladh, Luca Berardinelli, Hugo Bruneliere, Antonio Cicchetti, Mohammadhadi Dehghani, et al.. Towards Automating Model-Based Systems Engineering in Industry - An Experience Report. The 18th Annual IEEE International Systems Conference (SYSCON 2024), Apr 2024, Montreal, Canada. 10.1109/SysCon61195.2024.10553610 . hal-04448172v2

HAL Id: hal-04448172
https://hal.science/hal-04448172v2

Submitted on 16 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Towards Automating Model-Based Systems Engineering in Industry - An Experience Report

Johan Cederbladh*, Luca Berardinelli‡, Hugo Bruneliere§, Antonio Cicchetti*, MohammadHadi Dehghani‡, Claudio Di Sipio¶ , James Miranda§ , Abbas Rahimi‡ , Riccardo Rubei¶ , Jagadish Suryadevara†
*Mälardalen University, Västerås, Sweden – johan.cederbladh@mdu.se
†Volvo Construction Equipment, Eskilstuna, Sweden
‡Johannes Kepler University, Linz, Austria
§IMT Atlantique, LS2N (CNRS), Nantes, France
¶Università degli Studi dell'Aquila, L'Aquila, Italy

***Abstract*—Designing modern Cyber-Physical Systems (CPSs) is posing new challenges to both industrial practitioners and academics. In this context, adopting cutting-edge paradigms, such as Model-Based Systems Engineering (MBSE), DevOps, and Artificial Intelligence (AI), can offer new opportunities for improving CPS design automation. While such paradigms are already jointly used in the research community to support system design activities, there is a need to fill the gap between academia and industrial practitioners. Indeed, system specification is still mainly performed manually in many industrial projects. In this paper, we present a collaboration between industrial and academic partners of the AIDOaRt European project towards a model-based approach for CPS engineering applied in one of the project use cases. We identify key challenges and corresponding solutions to enhance the automation of CPS design processes. Notably, we consider a combination of prescriptive modeling, model transformations, model views, modeling process mining, and AI-based modeling recommendations. As an initial evaluation, the proposed approach is applied to a practical industrial case study.**

***Index Terms*—Model-Based Systems Engineering, Artificial Intelligence, DevOps, Cyber-Physical Systems**

I. Introduction

Cyber-Physical Systems (CPSs) integrate computation, networking, and physical processes with feedback loops where heterogeneous data are collected and analyzed [35]. With the advent of Industry 4.0, modern CPSs go beyond "just" computation and data collection. These complex systems are now widely adopted in multi-disciplinary domains, e.g., intelligent transportation systems [41], healthcare [21], and smart business [30]. Despite their advantages, their underlying complexity needs to be addressed during the whole CPS life cycle, i.e., from its specification to its actual deployment and operation. In particular, modeling CPSs from scratch is a challenging task since designers must handle several issues, including heterogeneity, concurrency, or time-based constraints [13]. In this respect, the integration of Model-Based Systems Engineering (MBSE) [33], [5], DevOps [26], and Artificial Intelligence (AI) [11] can come in handy to automate the design process.

In this paper, we present the current status of the collaboration started 2 years ago (and still ongoing) between different solution providers and use case providers in the context of the AIDOaRt research project [8]. Notably, the paper use case is one of several in the project, focusing on the digitalization of the construction equipment domain, pushing for the inclusion of cutting-edge technologies. From the practical industrial use case, we elicited a set of challenges and issues by relying on the experience of engineers in terms of interoperability, adoptability, and automation of the design activities.

As a result, we present an approach integrating solutions from several partners along with external (open-source) CASE tools. The proposed approach notably aims at leveraging MBSE and AI/ML capabilities for i) capitalizing on legacy engineering data, ii) supporting the structural modeling of CPS architectures and its variants via the SysML [20] and AutomationML [2] standards, and iii) allowing modeling recommendations via Machine Learning (ML) and process mining techniques. Our initial findings show that our approach can help to improve the automation of design operations, which are part of the typical engineering workflow. We plan to extend our approach further to support the continuous delivery and co-simulation of component variants [19] by integrating solutions for model-based DevOps and related standards.

The main contributions of the paper are the following: i) We propose an approach to integrate a modeling tool chain with an AI-based solution, ii) We present a preliminary evaluation of this approach on the challenges initially posed by engineers in the AIDOaRt use case, iii) We report on the feedback we collected from hackathons and a demo workshop with engineers, iv) We provide lessons learned and current limitations of the proposed approach for interested researchers and practitioners.

The rest of the paper is organized as follows. Section II introduces the background and related work. Section III presents the use case and corresponding challenges as the core

This work was partly funded by the AIDOaRt project, an ECSEL Joint Undertaking (JU) under grant agreement No. 101007350 in collaboration with Volvo Construction Equipment, the EU Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813884, and by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development (CDG)

motivation of our work. Section IV described our proposed approach as a solution to these challenges. Section V evaluates the results from the ongoing collaboration, hackathons, and a demo workshop. Section VI discusses lessons learned and limitations of the proposed approach (including possible mitigation actions). Section VII concludes the paper by opening on further possible improvements and future research.

II. Background and Related Work

Model-Based Systems Engineering for CPSs: MBSE is a paradigm where models are used as primary artifacts during all system engineering activities, in order to foster knowledge and data reusing, process automation, etc. [5]. Thus, MBSE practices provide suitable ways of tackling the complexity of engineering CPSs [37]. A recent systematic literature review [29] examined the past and current use of MBSE in a CPS context. While noting a high interest from both industry and academia, the authors observed a lack of empirical studies and concrete evaluations from industry. This paper, presenting our practical industrial experience of applying MBSE for CPS design, aims to make a step in this direction.

UML and SysML are the modeling languages considered the de facto standards for system modeling. Their widespread use is already well-documented [27], [40], and the interest in MBSE is constantly growing within various industrial contexts and domains. Additionally, previous industrial experiences [36], [4] have shown the potential of SysML and incited us to continue to developed its usage in the experiments presented in this paper. Similarly, AutomationML is a data format widely accepted in Industry 4.0 [2], [31]. It allows to represent general-purpose, hierarchical object information through Computer Aided Engineering Exchange (CAEX) [23]. Moreover, it comes with built-in linking mechanisms to 3D models using COLLAborative Design Activity (COLLADA) [25], and industrial control programs through PLCopen XML [22].

Model views as a federation tool: Beyond choosing modeling languages, engineering complex CPSs involves creating, transforming, and using multiple models describing various complementary system aspects [13]. Model views approaches provide unification mechanisms to federate and manipulate such heterogeneous models in a more transparent way [6]. Relying on such model-based principles and techniques allows specifying, creating, and handling views over models that possibly conform to different modeling languages. Once built, model views can be used to uniformly navigate, query, and transform the aggregated data from the various contributing models. In the context of AIDOaRt, we propose to rely on our scalable and computationally efficient approach to create and handle model views [7]. This approach comes with an EMF-based implementation called *EMF Views* [9], [7][1], a tool implemented on top of the Eclipse Modeling Framework (EMF)[2]. The use of this approach and related *EMF Views* tool has already been demonstrated in practice in the past within another CPS engineering context [17]. In this paper, we aim to further demonstrate its applicability as a relevant multi-view modeling solution for model federation in the context of the CPS engineering experiments.

Modeling event logs: Properly recording logs is of primary importance since systems can produce vast amounts of event data [38]. This requires efficient serialization methods to save storage space and memory footprint. Moreover, the data should be structured in a way that facilitates the analysis and mining of the recorded events. One of the most prominent standards is the IEEE Standard for eXtensible Event Stream (XES) [1], [24]. The XES standard provides an XML schema document that defines a generally acknowledged XML format to interchange event data between information systems (for the interoperability of event logs and event streams).

In this paper, we exploit a modeling event recorder (MER) leveraging XES-based logs and EMF-based technologies to generate datasets of modeling traces to feed AI-based strategies to support modeling activities.

AI-based strategies to support modeling activities: Supporting modeling activities is crucial for their acceptance by practitioners. Thus, Intelligent Modeling Assistants (IMAs) have recently attracted the interest of the MBSE community. We briefly detail some strategies to support modeling activities outside the industrial domain. Burgueño et al. [10] already proposed an architecture based on Natural Language Processing (NLP) for the auto-completion of partial models. Given a set of textual documents related to the initial model, relevant terms are extracted to train a contextual model using several NLP techniques. NEMO [15] supports the completion of BPMN models by exploiting the Long-Short Term Memory (LSTM) strategy. The approach encodes the modeling operations using a sequence-to-sequence decoder to predict the next modeling operation. Weyssow et al. [39] present a learning-based approach that exploits RoBERTa, a pre-trained neural network, to suggest relevant modeling language constructs [39]. The latter are first encoded as structured trees, then the RoBERTa model predicts the missing elements and provides the modeler with insightful domain concepts. In the context of this paper, we enrich the SysML and AutomationML modeling experience of engineers by exploiting the MORGAN tool [16]. The goal is to validate this tool in an industrial context by providing the user with relevant modeling operations that can be used to complete the model under construction.

III. Motivation

With the increased customer and regulatory emphasis on sustainability, the construction equipment domain is on a transformation journey focusing on the electrification of construction machines, including both battery-electric and fuel-cell technologies. Maintaining quality is paramount even during the transition period with fast prototyping and short lead times. This requires the application of new technologies, not only in the final product but also during development. In the collaboration through the AIDOaRt project, a set of use cases and challenges regarding the development of the mentioned

[1]https://www.atlanmod.org/emfviews/

[2]https://projects.eclipse.org/projects/modeling.emf.emf

future machines are presented. In this paper we use industrial use case in the form of a *Dumper System* (DS).

Figure 1a shows a high-level tool (boxes) landscape and practices and their inter-dependencies (lines) as currently used in practice for the software and system engineering of products and their variants in the AIDOaRt use case. The engineering process starts with office tools, e.g., Visio and Excel by Microsoft, which are extended with useful plugins (e.g., architecture description palettes with variability aspects) to produce requirements and architectural definitions. Then, experts manually inspect the resulting artifacts as guidance for coding software components and specifying simulation models of physical components of products. Variants offered by product lines are finally configured via change management tools.

Figure 1b shows the expected improvement from a tool landscape perspective thanks to the AIDOaRt project collaboration. New contributions brought by the AIDOaRt consortium (wavy boxes) are considered to foster the automation of the engineering process. MBSE techniques and practices are explicitly introduced, with modeling tools playing a key role in transforming descriptive engineering artifacts produced by office tools into models. The objective is to pave the way for integrating MBSE, AI/ML, and DevOps techniques and practices. Transforming current descriptive artifacts to a prescriptive model-based representation is a suitable step toward relieving and improving many bottlenecks in current processes. Starting from the perspective of a model-based approach for architecture descriptions, we identify a set of challenges from an industrial point of view:

CH1: Managing interoperability and traceability in the system development process: CPS engineering is a multi-disciplinary process and usually requires the integration of domain-specific languages and tools. Thus, interoperability is a major concern and can be realized by weaving techniques (e.g., model transformations) applied to artifacts produced at each stage of the system life-cycle. In this respect, traceability is another key indicator that tests the quality of the produced artifacts throughout the development process, i.e., from the requirements gathering to the actual development.

CH2: Promote the adoption of modeling practices in an industrial context: The adoption of modeling practices within industry is a common long-term challenge in the modeling community. Thus, it is naturally also reflected in this case study. To promote modeling in the engineering context, the proposed solution architecture needs to demonstrate the added value of modeling in this case. Consequently, throughout the paper, we discuss different challenges and lessons learned towards a clear path for user adoption.

CH3: Handling legacy artifacts: Legacy artifacts created by engineers are valuable assets and play an important role in the specification of new CPSs, especially since the current engineering methods, processes, and workflows rely on past expertise. On the one hand, a novel system must support the integration of the legacy artifacts to support the actual development of critical components. On the other hand, to be acceptable, new engineering practices must not disrupt well-established routines. Therefore, there is a need for a flexible solution capable of integrating both legacy and new approaches and artifacts.

CH4: Supporting automation through the combination of MBSE practices and AI-based tools: Engineers manually specify the system components through loosely integrated tools. Even though the engineering process is conducted correctly, MBSE practices can ease the burden of manual specification by offering a plethora of utilities, e.g., well-founded modeling environments, model transformations, and domain-specific languages. Furthermore, modeling activities can be automatized by employing AI-based algorithms. Therefore, adopting MBSE practices can improve the whole process by reducing the manual effort required by the engineers.

It is worth mentioning that the identified challenges aim to fulfill the AIDOaRt high-level goals [18] by relying on the existing tool landscape presented in Figure 1).

IV. Approach

Figure 2 shows the actors and use cases concerned in the context of our work. The main use case concerns the need to provide *system modeling* capabilities to industrial practitioners playing the role of *domain experts*. Existing office tools, like Visio and Excel, provide *descriptive system modeling* capabilities (Figure 2a). The current practice leverages Visio documents for *graphical representation* and Excel sheets for *variant descriptions*, both documenting product lines with components and their variants. The availability of industrial-grade APIs by office tools provides generic automation capabilities, resulting in complex implementations of customization plugins and poor automation results. To overcome this, the approach proposed by AIDOaRt partners aims to enable *prescriptive system modeling* capabilities (see Figure 2b).

Figure 3 depicts a solution architecture detailing the generic support for the landscape sketched in Figure 1b. It integrates partners' solutions, as offered to the whole AIDOaRt consortium, with open-source tools and newly developed components, like model transformations. The goal is to cope with specific automation and integration needs. Figure 4 depicts the engineering workflow supported by the proposed solution architecture in an activity-like diagram. The components' ids in Figure 3 are used to identify the corresponding supported engineering actions in Figure 4. The activities are organized in three swim lanes, each grouping actions and tools supporting modeling with SysML, AutomationML, and recommendation capabilities. It is worth noting that both SysML and AML modeling are meant to be interchangeable, allowing the import of legacy data and modeling recommendations.

The remainder of this section details the tool-supported activities shown in Figure 4. A demonstration video shows part of the engineering workflow in action, recovering engineering data from legacy Excel sheets[3]. In addition to the demonstration, we provide the produced materials, data sets, employed tools, and data related to the demo workshop.

[3]https://github.com/AIDOaRt-VCE-Team/Solution-framework

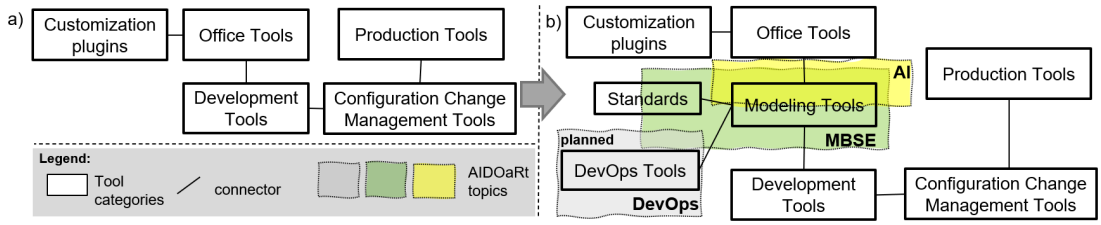


Fig. 1. Improving the automation of the engineering process as envisioned in AIDOaRt.

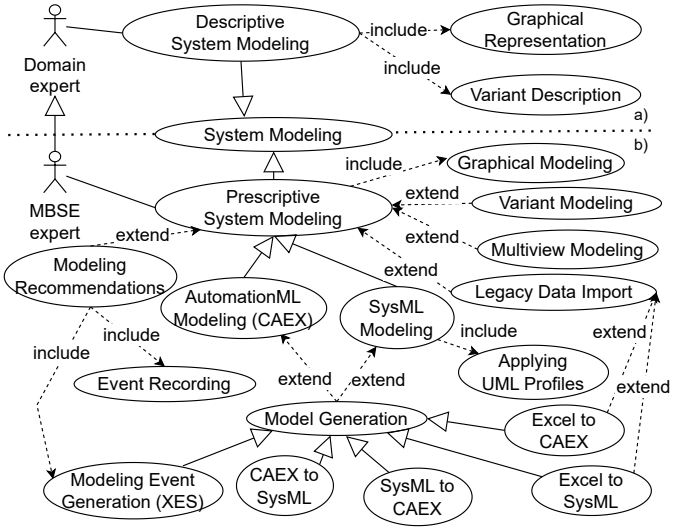


Fig. 2. Actors and use cases of our field of investigation.

A. Legacy data import (optional)

Legacy engineering data may be available in Visio and Excel documents. Visio is mainly employed to depict diagrams for human communication. Therefore, the proposed solution imports data only from Excel documents via model transformations (2a, 2b) implemented in Epsilon EOL[4], which provides ad-hoc Epsilon Model Connectivity (EMC) layers for Excel and EMF. EMF is the de-facto standard technology (a.k.a., meta-language [5]) used to specify the SysML and AutomationML standards' implementations within the Papyrus (3) and the CAEX Modeling Workbench (5) editors.

B. SysML modeling and AutomationML modeling

The current solution architecture and engineering workflow concentrate on modeling structural information of systems. For this purpose, SysML supplies block definition and internal block diagrams while the CAEX modeling workbench enables AutomationML modeling [28]. The latter is a Sirius-based[5] editor leveraging EMF-based technologies [3] and providing modeling support for CAEX models compliant with the CAEX standard version 2.15 and 3.0 (see Figure 6). Figure 5 shows an excerpt of the Dumper System in a SysML block definition diagram. It shows four blocks annotated with various stereotype applications, i.e., two `pull unit` variants with the `axle` and `water cooler` internal modules.

The canvas in Figure 6 shows an *instance hierarchy* of *internal elements* objects (IEs) (i.e., `dumper body`, `body attachment`, `water plate`, `tail light`, `tail gate`) suitably connected via *internal links* pairing mechanical *external interfaces* (`m` elements). The right-hand panel is a palette containing CAEX available modeling elements for editing the model.

C. Multiview modeling

As introduced earlier, EMF Views [9] is an approach and corresponding Eclipse/EMF-based tool that provides capabilities for specifying and building views over one or several models in potentially different modeling languages. In the presented work, EMF Views is used to support *Multiview Modeling* (Figure 2), conceived as an engineering activity independent from specific modeling languages. This way, it can be applied to federate any EMF-based models involved in the CPS engineering process (cf. Figure 4). Currently, it is already used to federate SysML and AutomationML models into integrated views that engineers can navigate and query depending on their needs (9). However, if required, other complementary models can be added in the future into these integrated views. Thanks to this, the engineers can more easily get an overall vision of the system under study and make design decisions accordingly without referring anymore to legacy data from Excel sheets.

D. Modeling event recording

The Modeling Events Recorder (MER) is a tool enabling the recording of user modeling actions [12]. It is an Eclipse plugin that interacts with Sirius-based graphical editors for EMF-based models, as the CAEX modeling workbench, and records users' modeling event logs. Event logs are encoded in XES, an XML schema dedicated to logs encoding. The MER tool also provides a XES implementation in EMF for serializing event logs as EMF-based models. With MER, the user interacts with the editor and creates, modifies, and deletes graphical elements from the modeling canvas like SysML blocks (see Figure 5) or AutomationML internal elements (see Figure 6). This change observation is done via the EMF Notification API, a built-in API in EMF. When the editing session is concluded[6], the plugin stores the collected traces in a XES

[4]https://eclipse.dev/epsilon/

[5]https://www.eclipse.org/sirius/

[6]A modeling session starts when the editor is opened and ends when the containing Eclipse project is closed.

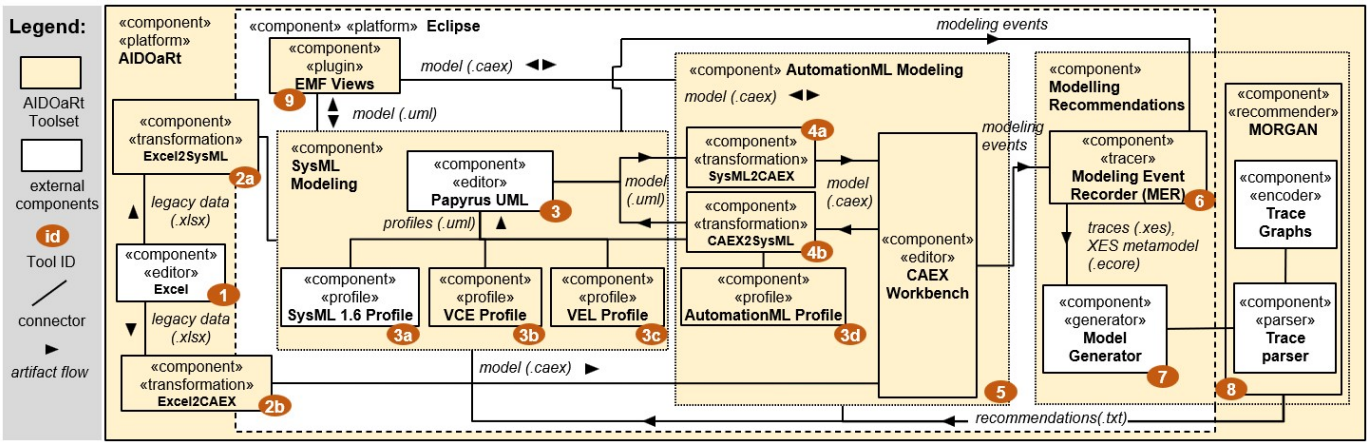


Fig. 3. The solution architecture for the use case challenges as part of the larger AIDOaRt framework.

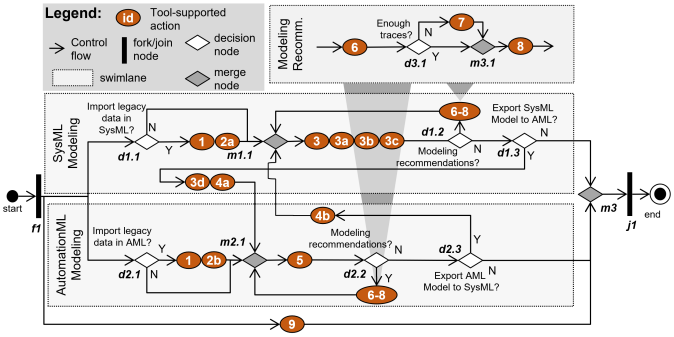


Fig. 4. The engineering workflow enabled by the solution architecture.

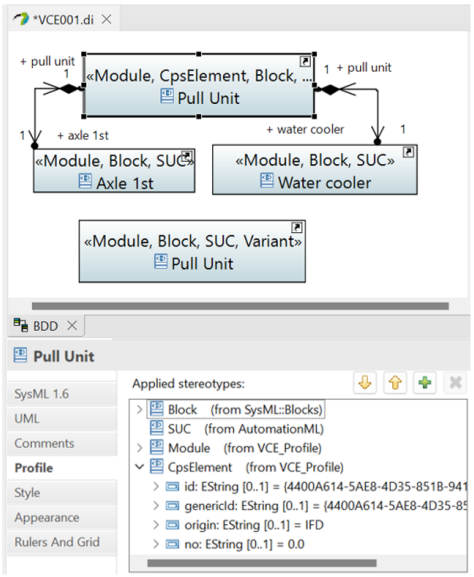


Fig. 5. Excerpt of an annotated SysML model of the DS.

log file. The MER output, i.e., XES-based traces, can be used with XES-compliant tools for different purposes (e.g., PRoM[7] for modeling process mining). In this work, MER's traces feed MORGAN to generate modeling recommendations.

E. Modeling recommendations

The *Modeling Recommendations* capability (Figure 2) is offered by the MORGAN tool. According to the engineering workflow in Figure 4, MORGAN (8) is trained with a data set of XES traces compliant with the EMF implementation of XES as brought by the MER tool. MER can generate them (6) when experts use SysML or AutomationML editors (3, 5) or by EMF-compliant model generators (7). Compared to the previous version [14], the current MORGAN's architecture introduces an *Trace parser* to extract relevant information from XES traces. This way, we obtain a textual-based representation used by the *Graph-encoder* to produce a list of trace graphs. To this end, the encoder extracts different features for each event, i.e., the type of event and the affected artifacts. Each graph is constructed using Natural Language Pre-processing (NLP) techniques, i.e., stemming, and dash-removal. Afterwards, we reuse the same underpinning algorithm presented in our previous work, i.e., the *Graph Kernel*. In particular, we assess the graph similarity between the XES graphs using the Weisfeiler-Lehman algorithm [34], provided by the Grakel Python library[8]. The mechanism is based on a pairwise comparison between two graphs (i.e., two models) according to the following formula:

$$k_{WL}(G, G') = k(G_0, G'_0) + k(G_1, G'_1) + \ldots + k(G_h, G'_h) \quad (1)$$

where h is the number of iterations, $G_0, G_1, ..G_h$ and $G'_0, G'_1, ..G'_h$ are the encoded sequences of the graph (i.e., model) G and G' respectively. The outcomes take the form of the ranked list of similar operations given the context of the modeler, i.e., the initial XES trace. As a result, MORGAN eventually produces a textual file containing the most relevant modeling operations to be performed.

[7]https://promtools.org/

[8]https://ysig.github.io/GraKeL/0.1a8/

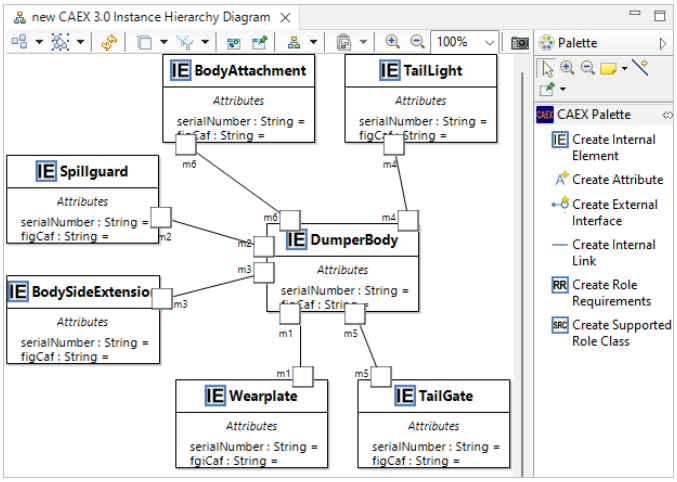


Fig. 6. Excerpt of AML model of the DS in the CAEX Modeling Workbench.

V. Evaluation

As a practical evaluation, we performed workshop sessions with engineers external to the development of the solution framework. Four engineers with different backgrounds participated to individual working sessions. We chose the participants in a way that we could get feedback potentially beneficial to our solution regarding different aspects. The sessions consisted of a quick introduction to the overall project, a hands-on demonstration of the current solution, and a discussion centered around a questionnaire to be filled out. The details of the workshop sessions are found on the GitHub page[9] and are summarized in Table I. Note that we considered a 1-5 Likert scale responses from the participants, where 1 corresponds to strongly disagree, 2 disagree, 3 indifferent/or similar to the current implementation, 4 agree, 5 strongly agree. ID1 refers to a Software Developer, ID2 refers to a Verification Engineer, ID3 refers to a System Architect, ID4 refers to a System Engineer. Based on these sessions and the provided feedback, we feel confident that the solution provides a promising added-value in the considered context. Our main takeaway is that the created models and added capabilities are seen as appropriate for the engineering activities. They also seem to be suitable in terms of complexity and usability for the involved engineers. Furthermore, we also evaluated our solution more generally regarding the challenges from Section III. This is supported by the activities performed during both the AIDOaRt hackathons and the Demo workshop providing a industrial perspective over our work.

Addressing CH1: The proposed solution architecture improves interoperability via standardized means of representing data. Indeed, using standard modeling languages paves the way for further connection to downstream activities, e.g., connections to simulation models in Simulink are expected in a near future. In particular, using system modeling languages such as SysML or AutomationML enables domain experts to unify artifacts via shared models. This also benefits to traceability, as necessary for the Product Line Engineering (PLE) context [32], and promotes a more vital link between the activities across the development.

Addressing CH2: The proposed solution promotes modeling by integrating additional features beyond just system modeling. The conducted workshop indicated that the engineers find the solution globally suitable for their engineering tasks. In particular, they all agree that the produced models are easy to understand and allow them to design the considered system properly. Furthermore, the solution can capture the information stored in the legacy artifacts even though the support of these legacy artifacts is not fully covered, e.g., the solution cannot produce the same Excel sheets as output. Altogether, the conducted survey demonstrates that the proposed solution can be a tool for promoting model-based practices with interesting capabilities compared to their current practices.

Addressing CH3: Handling legacy artifacts is the starting point for the current solution architecture. Indeed, the necessary data can be extracted from the currently used artifacts, thanks to text-to-model transformations, to populate SysML models. Moreover, the proposed solution is positioned to handle legacy artifacts both upstream and possibly downstream in the development flow. Thus, the solution integrates well into the current assortment of existing artifacts. In practice, the solution is not aimed to disrupt the existing approaches but rather to assist and improve the actual usage of existing artifacts. In this paper, we propose new complementary means for CPS development by introducing model-based representations of the engineered system.

Addressing CH4: Initial results have been extracted from the running of AIDOaRt hackathons. However, we believe that AI-based techniques are more globally relevant for industrial practitioners when modeling their CPSs. In this respect, analyzing the modeler's context can improve the overall quality of the recommended operations, as shown in the conducted demo workshop. The purpose of such recommendations is twofold: on the one hand, an expert modeler can increase its productivity; on the other hand, a junior developer can receive hints about the procedures. Due to more formalized means of modeling artifacts with standard languages, several components in the solution are fully automated. Several other components are already partially automated, and future ones can capitalize on the baseline we defined.

VI. Discussion

A practical gap between industry and academia. Different environments and contexts often create a gap in terms of needs and expectations. Thus, in our work, a significant effort has been and is still being made in order to foster the adoption of a solution coming from academia in the industrial context. The presence of legacy artifacts to consider, some limitations due to the company organization, different engineering activities to tackle, and various kinds of users to support are all challenges we also faced. To overcome this, it has been necessary to offer middle-ground paths for

[9]https://github.com/AIDOaRt-VCE-Team/Solution-framework

TABLE I
SUMMARY OF A SURVEY WITH ALL QUESTIONS AND ASSOCIATED LIKERT-SCALE SCORES.

Survey Question	Supported UC(s)	ID1	ID2	ID3	ID4	Avg.
1. Is the tool-chain usage suitable in complexity?	System Modeling	4	4	4	4	4
2. Is the support provided by the toolchain suitable for systems modeling?	System Modeling	4	5	3	4	4
3. Is the process easy to follow?	Descriptive System Modeling	4	3	5	4	4
4. Can the approach enhance the specification of the considered system?	Modeling recommendations	5	5	5	5	5
5. Are the added capabilities from the recommender system seen as suitable given the current context?	Modeling Recommendations	5	4	3	4	4
6. Is the added model federation and traceability capability beneficial for the modeling activities?	Multiview Modeling, Variant Modeling	3	5	5	5	4.5
7. Is AutomationML suitable for the modeling activities?	AutomationML Modeling	4	4	4	4	4
8. Is it easy to understand the models created?	Model Generation	5	5	5	5	5
9. Are the models seen as useful for engineering workflow?	Prescriptive System Modeling	4	4	4	5	4.25
10. Do the models capture cross-domain knowledge?	Multiview Modeling	4	3	3	4	3.5
11. Can the models accurately capture the information from Legacy artifacts?	Legacy Data Import	4	4	5	3	4
12. Does the methodology provide mechanisms to address the stakeholders' "concerns"?	Graphical Modeling	3	4	3	5	3.75
13. Is SysML a suitable language to be used as a companion with AutomationML?	SysML modeling, AutomationML modeling	3	3	5	5	4
Average score rounded to first decimal		4.0	4.1	4.1	4.4	4.2

collaboration, such as dedicated support for legacy artifacts. Moreover, we tried to follow a user-first approach and to stay generally driven by the problems and needs. Indeed, engineers are concerned with reusing as much as possible their legacy artifacts and embedded knowledge in order to limit the gap with their current practices. Another key reason for this situation is that the state-of-the-art in academia and the state-of-practice in industry are quite different. Something that is considered common knowledge or practice in academia can have a rather different reality in the industry. Concretely, matching the needs and finding common understanding in the context has been a significant challenge. Nevertheless, we were able to converge by working on realistic examples during the AIDOaRt hackathons, workshop, and demonstration sessions.

A proper pipeline to enable data-driven tools. In CPS engineering, the capability to efficiently automate activities and related tools mostly relies on the quality and relevance of the available data. As a consequence, data gathering and model completion are two critical tasks that are essential to improve the CPS engineering process when conducted properly. However, both the availability and quality of the required data cannot be systematically guaranteed. Actually, a pre-processing phase is very often needed to sufficiently prepare the data for further use. In the proposed solution, the MER component is capable of gathering the modeler's behavior by monitoring the Eclipse IDE. Although the collected data is already well-structured, it still must be pre-processed to be used for prediction purposes, thus enabling the model completion feature that we envision in the proposed architecture. In this respect, the feature extraction phase plays a crucial role in the predicting phase, as several hyper-parameters must be defined accordingly. The produced set of real XES traces from AutomationML and SysML modeling activities demonstrates that the employed AI-based component is promising and can already provide meaningful recommendations. In the next steps of our collaboration, we plan to gather, with the assistance of engineers, real data to be made compliant with the SysML and AutomationML standards.

Pitfalls when integrating heterogeneous modeling tools. The proposed solution intends to support the automation of different CPS engineering activities. To this end, several modeling tools need to be efficiently combined and to interoperate smoothly together. Unfortunately, this currently requires a rather deep knowledge of the involved modeling languages (general-purpose and/or domain-specific ones) as well as of the corresponding tooling support. Therefore, having a well-defined and unified central interface between the different interacting components appears to be necessary to support the aforementioned union of modeling languages and tools. In our solution, we opted for relying on the Eclipse/EMF environment as the common technical ground since it offers stable foundations and a set of consolidated technologies. However, each tool has its own characteristics (and constraints) in terms of required input data and output artifacts. Thus, this requires a development effort to *i)* conceptualize an integrated solution and *ii)* solve technical interoperability among the different tools. While *i)* has been addressed by using UML and related profiles as well as a model transformation chain, *ii)* still needs further integration work.

VII. CONCLUSION

To automate the engineering of CPSs, the AIDOaRt project aims to provide a methodology and framework that efficiently combine MBSE, AI, and DevOps tools. In this paper, we proposed a practical solution that combines such tools to support CPS engineering activities. We concretely evaluated and discussed this solution in the context of an industrial case study provided by the AIDOaRt research project.

In future work, we plan to cover additional scenarios by investigating on the integration of other solutions in our archi-

tecture and corresponding workflow, including AIDOaRt ones. For instance, we could integrate Simulink in order to cover the simulation and co-simulation of components from the engineering systems. To this end, we are already studying the Functional Mock-up Interface (FMI) standard as a way to assign different functional mock-up units to component variants specified in SysML models. Moreover, PLE is fundamental in the use case context as many systems share a common codebase or reusable components that can be specialized to meet various customer needs and operational scenarios. To support this, we plan to further investigate different variability modeling guidelines for SysML and AutomationML models. Finally, we could also study the possibility of adapting/deploying parts of the proposed solution architecture in the context of other AIDOaRt industrial case studies.

REFERENCES

[1] G. Acampora, A. Vitiello, B. Di Stefano, W. van der Aalst, C. Günther, and E. Verbeek, "IEEE 1849: The XES standard," *IEEE Computational Intelligence Magazine*, pp. 4–8, 2017.

[2] T. A. association, "Automationml," 2022, https://www.automationml.org/.

[3] L. Berardinelli, E. Maetzler, T. Mayerhofen, and M. Wimmer, "Integrating performance modeling in industrial automation through AutomationML and PMIF," in *INDIN 2016*, Jul. 2016, pp. 383–388.

[4] D. Bilic, E. Brosse, A. Sadovykh, D. Truscan, H. Bruneliere, and U. Ryssel, "An integrated model-based tool chain for managing variability in complex system design," in *MODELS 2019 Companion*. IEEE, 2019, pp. 288–293.

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

[6] H. Bruneliere, E. Burger, J. Cabot, and M. Wimmer, "A feature-based survey of model view approaches," *Software and Systems Modeling*, vol. 18, no. 3, pp. 1931–1952, Jun. 2019.

[7] H. Bruneliere, F. M. de Kerchove, G. Daniel, S. Madani, D. Kolovos, and J. Cabot, "Scalable model views over heterogeneous modeling technologies and resources," *Software and Systems Modeling*, vol. 19, no. 4, pp. 827–851, Jul. 2020.

[8] H. Bruneliere, V. Muttillo, R. Eramo, L. Berardinelli, A. Gómez, A. Bagnato, A. Sadovykh, and A. Cicchetti, "AIDOaRt: AI-augmented Automation for DevOps, a model-based framework for continuous development in Cyber–Physical Systems," *Microprocessors and Microsystems*, vol. 94, p. 104672, Oct. 2022.

[9] H. Bruneliere, J. G. Perez, M. Wimmer, and J. Cabot, "EMF Views: A view mechanism for integrating heterogeneous models," in *ER 2015*. Springer, 2015, pp. 317–325.

[10] L. Burgueño, R. Clarisó, S. Gérard, S. Li, and J. Cabot, "An nlp-based architecture for the autocompletion of partial domain models," in *CAiSE 2021*. Springer, 2021, pp. 91–106.

[11] Y. Dang, Q. Lin, and P. Huang, "Aiops: Real-world challenges and research innovations," in *ICSE 2019 Companion*, 2019, pp. 4–5.

[12] M. Dehghani, L. Berardinelli, and M. Wimmer, "Towards modeling process mining for graphical editors," in *Proceedings of the 26th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, 2020.

[13] P. Derler, E. A. Lee, and A. S. Vincentelli, "Modeling cyber–physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, 2011.

[14] J. Di Rocco, C. Di Sipio, D. Di Ruscio, and P. T. Nguyen, "A gnn-based recommender system to assist the specification of metamodels and models," in *MODELS 2021*, 2021, pp. 70–81.

[15] J. Di Rocco, C. Di Sipio, P. T. Nguyen, D. Di Ruscio, and A. Pierantonio, "Finding with nemo: A recommender system to forecast the next modeling operations," in *MODELS 2022*, 2022, p. 154–164.

[16] C. Di Sipio, J. Di Rocco, D. Di Ruscio, and P. T. Nguyen, "MORGAN: a modeling recommender system based on graph kernel," *Software and Systems Modeling*, Apr. 2023. [Online]. Available: https://doi.org/10.1007/s10270-023-01102-8

[17] R. Eramo, F. Marchand de Kerchove, M. Colange, M. Tucci, J. Ouy, H. Bruneliere, and D. Di Ruscio, "Model-driven Design-Runtime Interaction in Safety Critical System Development: an Experience Report." *The Journal of Object Technology*, vol. 18, no. 2, pp. 1:1–22, 2019.

[18] R. Eramo, V. Muttillo, L. Berardinelli, H. Bruneliere, A. Gomez, A. Bagnato, A. Sadovykh, and A. Cicchetti, "Aidoart: Ai-augmented automation for devops, a model-based framework for continuous development in cyber-physical systems," in *DSD 2021*, 2021, pp. 303–310.

[19] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–33, 2018.

[20] T. O. M. Group, *SysML*, 2019, https://sysml.org/sysml-specs/.

[21] S. A. Haque, S. M. Aziz, and M. Rahman, "Review of cyber-physical system in healthcare," *International Journal of Distributed Sensor Networks*, vol. 10, no. 4, p. 217415, 2014.

[22] IEC-61131-10, "Plc open xml exchange format," 2019, https://plcopen.org/iec-61131-10.

[23] IEC-62424, "CAEX," 2007, https://www.plt.rwth-aachen.de/cms/PLT/Forschung/Projekte2/~ejwy/CAEX_IEC_62424/?lidx=1.

[24] IEEE, "1849-2016 xes standard," https://www.xes-standard.org/, Aug. 2021, accessed: 2022-03-03.

[25] ISO17506, "Collada - iso 17506:2022," 2022, https://www.iso.org/standard/78834.html, last accessed on 30.11.2022.

[26] R. Jabbari, N. bin Ali, K. Petersen, and B. Tanveer, "What is devops? a systematic mapping study on definitions and practices," in *XP2016*, ser. XP '16 Workshops, 2016, pp. 1–11.

[27] I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang, "What industry needs from architectural languages: A survey," *IEEE Transactions on Software Engineering*, vol. 39, no. 6, pp. 869–891, 2013.

[28] T. Mayerhofer, M. Wimmer, L. Berardinelli, and R. Drath, "A model-driven engineering workbench for caex supporting language customization and evolution," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 2770–2779, Jun. 2018.

[29] M. A. Mohamed, G. Kardas, and M. Challenger, "Model-driven engineering tools and languages for cyber-physical systems–a systematic literature review," *IEEE Access*, vol. 9, pp. 48 605–48 630, 2021.

[30] R. A. Nieto, "Data fusion for automated business decisions," Aug. 25 2009, uS Patent 7,580,878.

[31] Platform Industrie 4.0, "The Asset Administration Shell: Implementing digital twins for use in Industrie 4.0," https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/VWSiD%20V2.0.html, 2019, accessed: 2022-12-01.

[32] K. Pohl, G. Böckle, and F. Van Der Linden, *Software product line engineering*. Springer, 2005, vol. 10.

[33] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 101–111, 2011.

[34] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *J. Mach. Learn. Res.*, vol. 12, p. 2539–2561, Nov. 2011.

[35] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of Cyber-Physical Systems," in *WCSP 2011*, Nov. 2011, pp. 1–6.

[36] J. Suryadevara and S. Tiwari, "Adopting mbse in construction equipment industry: An experience report," in *APSEC 2018*. IEEE, 2018, pp. 512–521.

[37] H. Thompson, M. Reimann, D. Ramos-Hernandez, S. Bageritz, A. Brunet, C. Robinson, B. Sautter, J. Linzbach, H. Pfeifer, V. Aravantinos *et al.*, *Platforms4CPS, Key Outcomes and Recommendations*. Steinbeis, 2018.

[38] W. van der Aalst, *Process Mining*. Springer, 2016.

[39] M. Weyssow, H. Sahraoui, and E. Syriani, "Recommending metamodel concepts during modeling activities with pre-trained language models," *Software and Systems Modeling*, vol. 21, no. 3, pp. 1071–1089, Jun. 2022.

[40] S. Wolny, A. Mazak, C. Carpella, V. Geist, and M. Wimmer, "Thirteen years of sysml: a systematic mapping study," *Software and Systems Modeling*, vol. 19, no. 1, pp. 111–169, 2020.

[41] G. Xiong, F. Zhu, X. Liu, X. Dong, W. Huang, S. Chen, and K. Zhao, "Cyber-physical-social system in intelligent transportation," *IEEE/CAA Journal of Automatica Sinica*, vol. 2, no. 3, pp. 320–333, 2015.