Asset Administration Shell as a Backbone for Interoperable and Scalable Discrete-Event Simulation

Ana Beatriz Fontão, Romão Santos, João Ruivo
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
ana.b.fontao@inesctec.pt

Angelos Bavelos
Netcompany
2b, rue Nicolas Bové
L-1253 Luxembourg

Dimosthenis Dimosthenopoulos
LMS
University of Patras
Patras 26504, Greece

Wallen Ribeiro, António Lucas Soares, António Baptista
INESC TEC, FEUP
Rua Dr. Roberto Frias
4200-465 Porto, Portugal

KEYWORDS

Asset Administration Shell; Discrete Event Simulation; Interoperability; Scalability; FlexSim

ABSTRACT

In the context of Industry 4.0, the concept of Asset Administration Shell (AAS) has emerged as a standardized digital representation of assets. As manufacturing environments become increasingly complex, it becomes crucial to ensure interoperable and scalable data exchange throughout the entire life-cycle of its assets. Simulation models, implemented by discrete event simulation tools (e.g. FlexSim), play a critical role on the planning, validation and optimization of manufacturing systems. However, their integration with standardized digital asset representations remains limited. This paper proposes a methodology for the integration of AAS instances with simulation tools to enhance the communication between physical assets, their digital representations, and simulation environments. This approach relies on standardized AAS models to enable consistent and efficient data exchange, supporting seamless integration between systems and remaining effective as implementations scale.

INTRODUCTION

The rapid evolution of smart manufacturing systems has made it crucial to seamlessly integrate numerous components into a compatible and efficient production environment. For instance, Digital Twins (DT), now becoming the backbone of cyber-physical systems management, heavily rely on heterogeneous data sources (Zhang et al. 2025). In this context, the Asset Administration Shell (AAS) emerged as a digital representation of an asset, which may be a physical, a digital or a logical entity (Alexopoulos et al. 2025). It enhances syntactical and semantical interoperability between systems through the use of a standardized structure in the definition of digital models (Heppner et al. 2023). The AAS of an asset may store data related to its properties, its capabilities and even its status in real-time (Alexopoulos et al. 2025).

Moreover, smart manufacturing environments increasingly depend on advanced decision-support tools to manage their growing complexity and variability. As a result of these unpredictable conditions, organizations start to rely on simulation tools as enablers for the analysis, prediction and optimization of production systems, without having to implement changes in the physical system. By reproducing real production conditions in a virtual environment, simulation software supports informed decision-making, accelerating production cycles and reducing both costs, and operational risks (Isania and Casalino 2025, Gunalan et al. 2026).

Despite the proven effectiveness of simulation tools in the aforementioned functionalities, these technologies still face challenges regarding interoperability and scalability in smart manufacturing systems (Zhao et al. 2024a). While several neutral data exchange standards have been proposed to enable multi-level data exchange and automated model generation, there is no comprehensive data model capable of supporting full-scope interoperability across simulation tools and manufacturing systems (Zhao et al. 2024a). Additionally, simulation platforms demonstrate strong capabilities in modeling and optimizing specific value streams, but their performance often declines when models are scaled to multiple lines, factories, or enterprise-level decision-making contexts (Isania and Casalino 2025).

To address these challenges, this paper explores the integration of AASs with discrete event simulation tools, enabling a standardized and interoperable data exchange between manufacturing assets or services and simulation environments.

This paper is structured as follows: The first section reviews relevant background and related work on AAS and simulation interoperability. The following section presents the proposed integration approach between AAS data model and simulation tools. The third section highlights the main findings from this work and the final section outlines directions for future research.

BACKGROUND AND RELATED WORK

This section presents the relevant concepts and related work to this study. It introduces the Asset Administration Shell and its role in standardized data exchange, reviews discrete event simulation in smart manufacturing environments, and discusses existing approaches that combine Digital Twins, AAS, and simulation-based engineering.

Asset Administration Shell

Initially proposed as part of the Plattform Industrie 4.0 initiative in Germany as the digital representation of a physical asset in the RAMI 4.0 architecture, the AAS was developed with the intent of standardizing how assets exchange data and services for Industry 4.0 interoperability (Sakurada et al. 2025). The AAS is a standardized digital representation of an asset in Industry 4.0, designed to provide uniform access to an asset's information and services across tools and organizations (Havur et al. 2025, Pajpach et al. 2025). Since Industry 4.0 architectures require heterogeneous systems, AAS is often described as a key interoperability enabler because it provides a standard structure and standardized interfaces to access asset data, supporting both file-based exchange and API-based interaction (Pajpach et al. 2025, Kaya et al. 2025). Especially, in DT systems, this is relevant to keep datasets consistent, addressable and exchangeable across the asset's entire lifecycle (Havur et al. 2025, Zhang et al. 2025).

From an Industry 4.0 perspective, interoperability is the capability of heterogeneous cyber-physical systems to exchange information/services and to use what is exchanged effectively, and is typically divided into two dimensions: syntactic interoperability and semantic interoperability (Huang et al. 2021). The first one is related to the systems' capability of technically exchange data, for instance through standardized protocols. The latter represents the system's ability to understand the meaning of external data (e.g. identifiers) (Havur et al. 2025, Huang et al. 2021).

Conceptually, an AAS is a container with a unique identifier that organizes all relevant information about an asset into modular parts called submodels (Pajpach et al. 2025). A submodel represents the aggregation of characteristics of an asset. Within each submodel, there are submodel elements, which hold values, references or even express relationships with other entities (Pajpach et al. 2025, Huang et al. 2021). Currently, there are several submodel templates approved and published by the Industrial Digital Twin Association (IDTA) in their GitHub repository[1]. It is highly recommended to use these templates as they provide standardized fields, which emphasize semantic interoperability. Additionally, a key mechanism here is the Data Specification IEC 61360 template, which aligns AAS concept definitions with the IEC 61360 data model (Kaya et al. 2025).

There are two kinds of AAS: template and instance. A template refers to the definition of what an AAS of a certain asset type must contain, for example, required fields, specific semantics and submodel structure. Then, based on the template, concrete AAS and submodel instances are created and populated with values related to a specific asset (Pajpach et al. 2025).

AAS information can be exchanged in multiple standardized ways, heavily contributing to syntactic interoperability. Firstly, focusing on file-based storage, there is the AASX package (Industrial Digital Twin Association - IDTA) 2024), which is a generic package file format specially tailored to the AAS structure. This kind of file can be created, updated and viewed using the AASX Package Explorer[2] tool. Then, it is also possible to store AAS in serialized file formats, namely JSON(JavaScript Object Notation) and XML(Extensible Markup Language). In this case, the JSON format is generally preferable as it is commonly used to cleanly map HTTP payloads, useful in REST-based implementations (Pajpach et al. 2025, Kaya et al. 2025). Lastly, from an implementation standpoint, Eclipse BaSyx[3] is an open-source middleware specifically built to manage AAS data through the use of HTTP/REST interfaces. BaSyx also provides SDKs, for example, to integrate with Python and Java applications, which support practical implementations of AAS-based environments.

DES in Smart Manufacturing

The growing need for flexible and smart manufacturing processes has emphasized the role of simulation — particularly Discrete Event Simulation (DES) — as a core decision-support approach for designing, evaluating, and improving production systems (Schutz et al. 2025, De Felice et al. 2025). DES is a widely used methodology, characterized by its discrete nature, meaning that the model's state changes as events occur and is therefore represented by a series of specific occurrences connected to a point in time (Schutz et al. 2025). DES tools are key enablers of DT systems, as they can act as a bridge between the physical and digital environments, enabling scenario testing without affecting the physical system (Santos et al. 2024). In fact, they are mainly used to test hypothetical scenarios, optimize processes, analyze performance, identify bottlenecks, and predict

[1] `https://github.com/admin-shell-io/submodel-templates` accessed on 28/01/2026

[2] `https://github.com/admin-shell-io/aasx-package-explorer` accessed on 28/01/2026

[3] `https://basyx.org/` accessed on 28/01/2026

the behavior of complex systems (Isania and Casalino 2025, Schutz et al. 2025). Organizations benefit from these capabilities to efficiently manage the performance of their systems, reduce implementation risks and costs, and to be more flexible when faced with increasing market demands and variability (Monek and Fischer 2023, De Felice et al. 2025).

Despite the clear advantages of simulation tools, they also present some challenges, namely standardization, interoperability and scalability. Firstly, a recurring challenge is the lack of a comprehensive data model. Although there are multiple industrial standards and standard data exchange standards, the literature suggests that semantic interoperability is still not achieved in DES tools (Zhao et al. 2024a). Secondly, interoperability is a major struggle in smart manufacturing simulation, mainly because these models must be able to connect with diversified systems across organizations (e.g. at Supply Chain level) and shop-floor systems (Zhao et al. 2024a). Lastly, the dynamic and complex characteristics of smart manufacturing environments create higher scalability demands; simulation models must remain reliable and efficient while also being able to receive larger volumes of data. This issue also relates to the topics previously mentioned, regarding heterogeneous systems and data sources. DES is increasingly integrated with real-time data and IoT, raising challenges for scaling from isolated models to broader factory and Supply Chain (SC) digital representations (Monek and Fischer 2023, De Felice et al. 2025).

The Path for Standardization - DES Tools and Integration Challenges

Commercial DES tools are frequently used to implement the simulation layer in Indstry 4.0 manufacturing systems. As previously mentioned, within DT-based systems, simulation tools play a key role in connecting the physical and digital domains, allowing for replication, analysis and optimization of system behavior across different decision-making levels (Santos et al. 2024). Among these tools, FlexSim[4] is the one that will be used in this study. FlexSim provides a powerful simulation environment for manufacturing, supporting 3D modeling of production systems and the ability to run experimental scenarios (Isania and Casalino 2025). The tool has been widely applied in manufacturing contexts to visualize production dynamics, evaluate workflow efficiency, and optimize resource allocation before the real-world implementation, contributing to reduced costs and accelerated production cycles (Lewicki et al. 2024). This program offers the ability to connect to external systems, including the ability to execute external code using FlexScript[5]. This versatility is essential in Industry 4.0 environments, for instance, to perform the integration of simulation models with various data sources and software components (e.g. real-time data exchange via IoT infrastructures) (De Felice et al. 2025, Monek and Fischer 2023).

Despite these capabilities, the integration of simulation tools with manufacturing systems is mainly domain-specific, including manual data input, file-based data exchanges and custom APIs. Some studies report that integration techniques often rely on very specific and proprietary data models, leading to limited interoperability and difficulties when integrating simulations with other systems (Krenczyk et al. 2018, Zhao et al. 2024a). In Zhao et al. (2024b), the authors highlight that, although there are multiple industrial standardized data formats exists, a comprehensive data model capable of supporting full-scope interoperability in DES is yet to be established. Another evidence of these challenges is shown in Shi et al. (2026), where the authors explain how existing DT implementations often struggle from limited semantic consistency and scalability, as data is manually configured and tightly coupled to specific tools or applications.

Overall, the literature highlights that simulation models often remain isolated, difficult to reuse, and challenging to scale beyond a certain scope. This research gap extends to the absence of standardized asset representations, semantic consistency, and reliable integration mechanisms for DES tools in smart manufacturing systems.

PROPOSED APPROACH

This work proposes an AAS-based integration approach to connect external decision-support applications – such as warehouse management systems, real-time monitoring systems or optimization algorithms – with FlexSim discrete event simulation in a modular and reusable way. Although this study focuses on the FlexSim software, it aims to be replicable with other DES tools. The core idea is to use AAS models as a standardized digital representation for simulation scenarios data exchange (parameters and results), decoupling simulation models from tool specific data formats. The following section presents some brief guidelines for the definition of AAS models, along with an example of an AAS model of a supply chain structure. After that, the proposed system architecture is introduced, including some technology suggestions. The third section shows a sequence diagram showing the flow of data and the role of each entity of the system in this process.

[4] `https://www.flexsim.com/` accessed on 28/01/2026

[5] `https://docs.flexsim.com/en/24.1/Reference/DeveloperAdvancedUser/ConnectingToExternalCode/ConnectingToExternalCode.html` accessed on 28/01/2026

AAS and Semantics Definition

The first step to define an AAS model is to prepare templates of the data, which define the characteristics of the assets that will be represented. In practice, the main distinctions between an AAS template and instance is the assetKind property – assigned *type* in templates and *instance* for instances. Similarly, the submodels have a field, named *kind*, which can be of value *template* or *instance*. Additionally, the two kinds of submodels differ since the instances' properties and fields are assigned value, while the templates only specify the type of properties. The creation of AAS and submodel instances can be done through the BaSyx's registry feature using APIs.
As previously mentioned, in the AAS context it is heavily encouraged to, when possible, use template submodels published and approved by the IDTA, along with well defined industry-specific standards. The AAS environment's semantic interoperabilty is improved by referencing existing semantics – using the *SemanticId* field – or by locally creating semantics as *ConceptDescriptions* with unique identifiers (Industrial Digital Twin Association - IDTA) 2023). By grounding submodel elements in established semantic references, the AAS ensures that exchanged data is not only machine-readable but also machine-understandable, which is essential for semantic interoperability across heterogeneous tools and organizations.
Figure 1 illustrated the AAS template created represent a Site. The BaSyx AAS UI shows, on the left, the AASs that describe the Supply Chain, Sites, Transports and Transportation Tasks. For the selected Site AAS, multiple submodels are available, describing and structuring the information required to characterize the asset and its connection with the other AASs. To ensure traceability and semantic consistency, these models: (i) have consistent identifiers (same structure for every AAS and submodel); (ii) are defined using submodel templates (e.g. Asset Location); (iii) and are associated with locally defined semantics, such as Coordinated Universal Time (UTC) associated with timestamps in the Schedule submodel.

Proposed Architecture

The architecture diagram shown in figure 2, presents the high-level design of a DT-based system with focus on the AAS and simulation layers.
Firstly, the **AAS management layer** is implemented using an Eclipse BaSyx **AAS server/registry** coupled with a **non-relational database** (using MongoDB) for storage. In this context, the use of MongoDB is relevant since it provides flexible and schemaless data persistence for various objects in JSON format, including AAS instances, templates, and submodels. The **external applications** module represents other systems of the DT

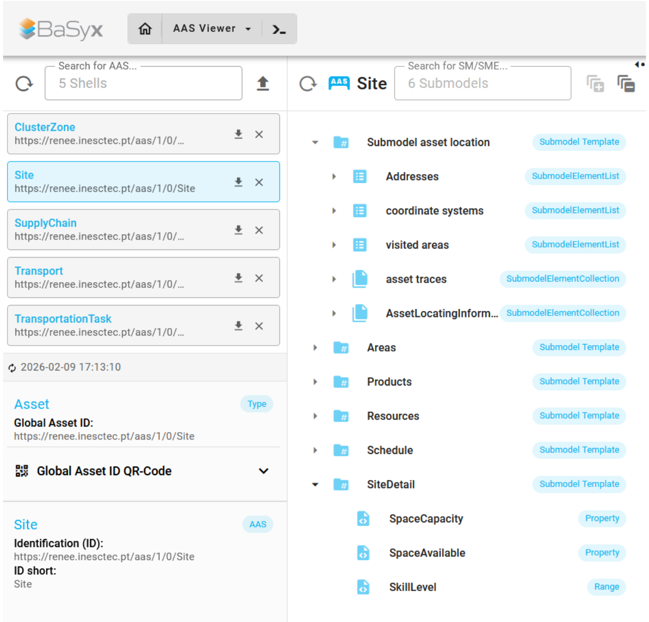

Figure 1: Template for the Site AAS on the Eclipse BaSyx UI

environment, including the application frontend, backend, a relational database and business logic. The connection between this component and the AAS server allows for CRUD operations, both relevant for simulation-related tasks, and for other DT functionalities like monitoring.
The **simulation adapter service**, implemented in Python and seen in the intersection of the simulation and AAS layers, serves as a bridge between the AAS representation and FlexSim's execution requirements. This component has two main responsibilities: (i) AAS CRUD operations, including retrieval of instances for scenario parameters, and insertion or update of instances according to simulation results; and (ii) conversion from AAS JSON structures into FlexSim-compatible data structures. This service is implemented to be flexible and adaptable to any AAS and submodels mapping.
Then, on the **simulation layer** side, the FlexSim **simulation software** is the core platform responsible for running simulations. In combination with this module, the proposed architecture presents a **FlexSim integration script** implemented with FlexScript, that acts as the interface between the simulation model and external code, namely, the simulation adapter service. Moreover, in the diagram it is possible to see a connection between the FlexSim component and external applications, namely the backend. This is the channel that triggers simulation runs and transports their results back to the DT core business logic component so they can be processed.
The goal of combining the AAS standardized digital rep-

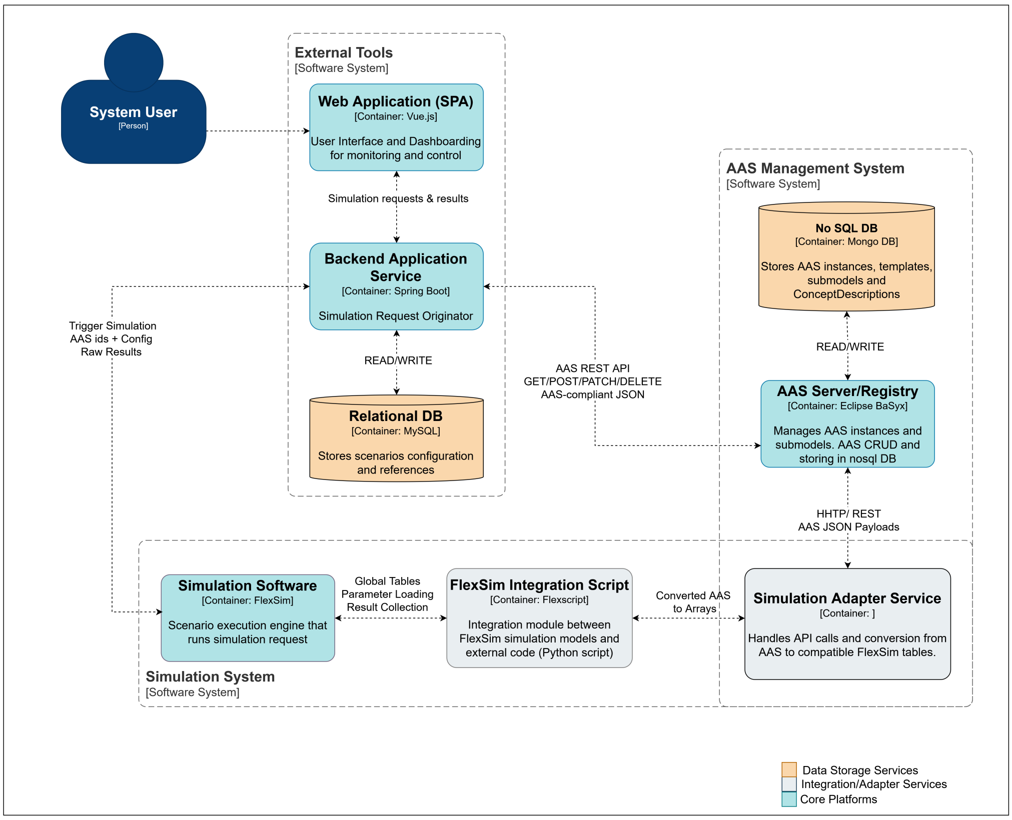

Figure 2: Architecture diagram for the proposed approach

resentation, persistent information support, and a independent simulation-based layer with flexible adapters is to reduce integration efforts, while also improving interoperability between tools and system scalability in complex heterogeneous environments that deal with simulation tasks.

Expected System Interaction And Data Flow

Figure 3 describes the expected interaction between systems and specifies the data flow of the proposed approach.

The execution starts when the external applications submit a simulation scenario request – **Phase 1**. The user initiates this call by specifying the required parameters using the system's user interface. This information is then passed to the application's backend, that is responsible for identifying the relevant AASs for the simulation run and passing them to FlexSim. Although this initial data exchange is essential, it is omitted from the sequence diagram to reduce complexity and keep the focus on the proposed AAS–FlexSim integration. Upon receiving the request, FlexSim uses the integration script (FlexScript) to retrieve the necessary input data. In this stage – **Phase 2** –, this script's main purpose is to act as a bridge between the simulation model and external services, namely the simulation adapter services (service in Python). At this point, this module loops through the list of required AASs and submodels, and retrieves them from the database using standardized HTTP requests to the AAS Server.

In **Phase 3**, responsible for the conversion of AAS to arrays, the simulation adapter first aggregates the retrieved AAS and submodel payloads into a single AAS-compliant JSON object. This step is crucial because FlexSim does not consume AAS semantic structures (either in JSON or .aasx) as parameters. The adapter then converts the AAS and submodels JSON into an array, which are then passed as a response to the FlexScript service – starting **Phase 4**. The script receives the converted parameter arrays and maps them to FlexSim's internal data structures, by populating tables and setting the corresponding parameters. Once the simulation software has the correct configuration, the simulation model can execute the run.

Phase 5 begins at the end of the simulation run, where FlexSim produces raw results, that may include KPIs, event logs or output tables, depending on the simulation model and its configuration. These simulation outputs are passed to the application's backend, where they are parsed and structured into AAS data, with the predefined JSON configuration. Moreover, this step is responsible for the data storage in the persistence database. Similarly to the loop block of phase 2, the insertion

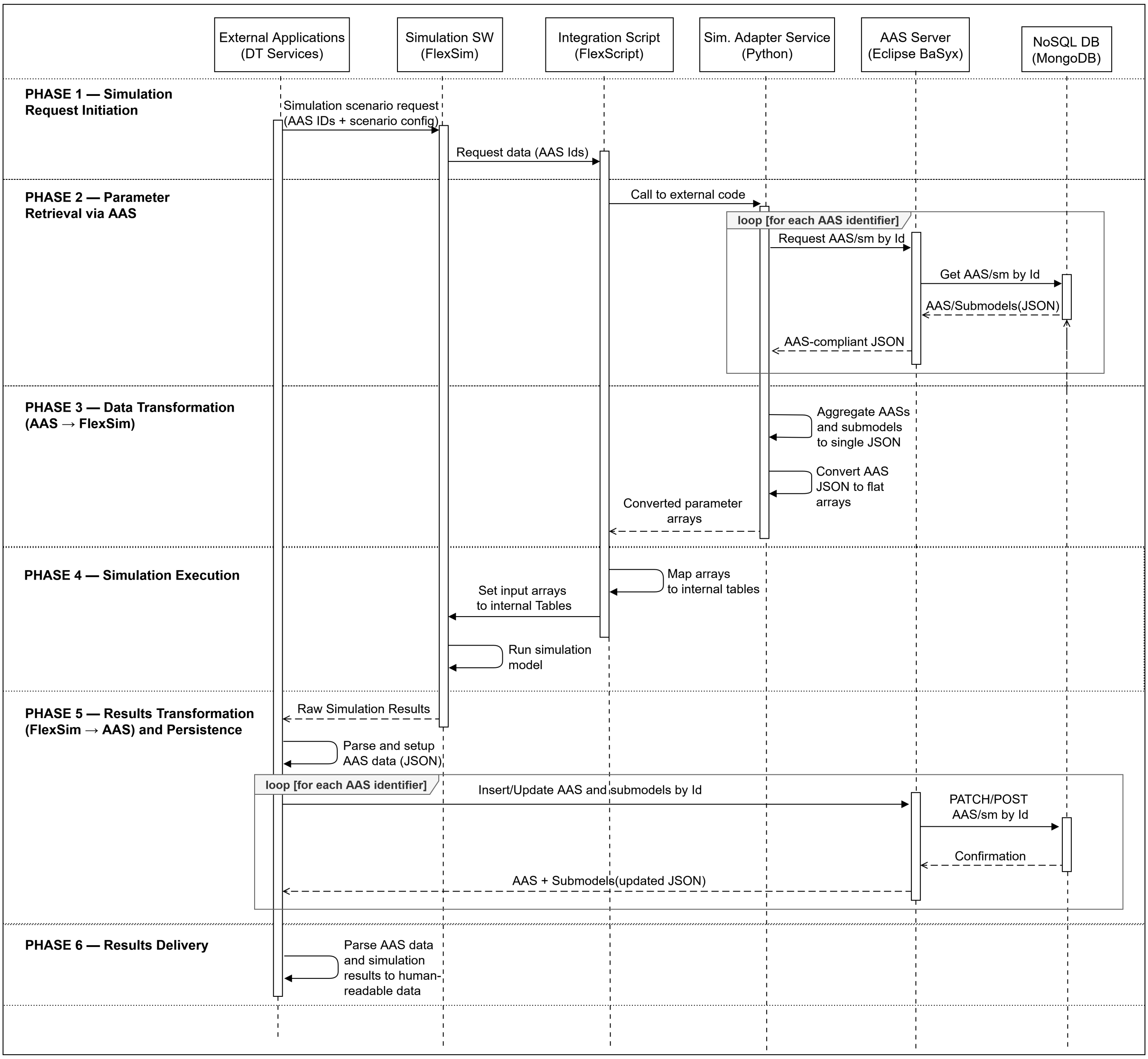

Figure 3: Sequence Diagram for the proposed approach

of AAS and submodels resultant from the simulation is done sequentially by issuing standardized HTTP requests to the BaSyx AAS server, which inserts or updates (based on the need) in the MondoDB database.
Finally, **Phase 6** is in charge of delivering the simulation results to the user, and closes the expected flow of the proposed approach Here, the raw AASs are translated into a human-readable format suitable for visualization and analysis in the user interface.
Overall, the goal of this sequence is to create a repeatable and reliable way to process simulation requests and results using standardized data structures. By clearly separating the different steps in independent components, the workflow can remain interoperable across diverse systems and easier to extend as new data or simulation scenarios are introduced.

CONCLUSION

This work presents an AAS-based approach to integrate standardized digital asset representations with FlexSim discrete-event simulation, aiming to improve interoperability and scalability in heterogeneous Industry 4.0 environments. Regarding technologies, the proposed architecture uses Eclipse BaSyx and MongoDB to manage and store AASs, submodels and concept descriptions, along with a Python-based simulation adapter to connect with FlexSim. Furthermore, the presented data flow demonstrates how AAS can be integrated with the simulation software in complex systems. This sequence of simulation scenario definition, execution, and result management, enables external applications to execute simulations and consume outcomes through a standardized interface.

FUTURE RESEARCH

Given the conceptual nature of this work, the next steps will focus on the implementation of the proposed approach in a real case study. It will also be applied in a broad scope, including supply chain and shop floor assets, addressing the data and interoperability gaps between them. As previously mentioned, the AAS models representing these components are already implemented and will be leveraged in the DT-based system to store, monitor and simulate hypothetical scenarios.

Validation will be extended to more realistic scenarios, assessing scalability, interoperability across heterogeneous tools via semantic AAS interfaces, and maintainability through modular evolution over time. The architecture will also be enhanced with AI multi-agent decision support and a natural-language layer, leveraging AAS semantics to improve grounding, transparency, and traceability.

In summary, these next steps will consolidate the proposed architecture into a robust foundation for interoperable simulation while enabling intelligent, agent-assisted decision support built on standardized AAS interfaces.

ACKNOWLEDGMENTS

This work was supported by the European Union's Horizon Europe research and innovation program through the RENÉE project (grant no. 101138415).

REFERENCES

Alexopoulos A.; Kalogeras G.; Koutras K.; and Kalogeras A., 2025. *Why Asset Administration Shells: A Survey on Uses and Challenges*. *IEEE Access*, 13, 126582–126609. ISSN 2169-3536. doi:10.1109/ACCESS.2025.3589931. URL https://ieeexplore.ieee.org/document/11082140/.

De Felice F.; De Luca C.; Petrillo A.; Forcina A.; Ortiz Barrios M.A.; and Baffo I., 2025. *The Role of Digital Transformation in Manufacturing: Discrete Event Simulation to Reshape Industrial Landscapes*. *Applied Sciences*, 15, no. 11, 6140. ISSN 2076-3417. doi:10.3390/app15116140. URL https://www.mdpi.com/2076-3417/15/11/6140.

Gunalan A.K.; Naik S.; Yoon S.; Wang Z.; and Ortell S., 2026. *Simulation-Driven Decision-Making for Process Optimization for Gate Valve Value Stream Production Line*. In K. Srihari; M.T. Khasawneh; S. Yoon; and D. Won (Eds.), *Flexible Automation and Intelligent Manufacturing: The Future of Automation and Manufacturing: Intelligence, Agility, and Sustainability*. Springer Nature Switzerland, Cham. ISBN 978-3-032-07675-5, 170–186.

Havur G.; Pellegrini T.; Schenner G.; and Fusco F., 2025. *A Design Study for Interoperable Digital Product Passports: Embedding UNTP Semantics in AAS Submodels*.

Heppner S.; Miny T.; Kleinert T.; Becker M.; Schmitz K.; and Alt R., 2023. *Asset Administration Shells as Data Layer for Enabling Automated Simulation-based Engineering*. In *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, Sinaia, Romania. ISBN 979-8-3503-3991-8, 1–7. doi:10.1109/ETFA54631.2023.10275474. URL https://ieeexplore.ieee.org/document/10275474/.

Huang Y.; Dhouib S.; and Malenfant J., 2021. *An AAS Modeling Tool for Capability-Based Engineering of Flexible Production Lines*. In *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Toronto, ON, Canada. ISBN 978-1-6654-3554-3, 1–6. doi:10.1109/IECON48115.2021.9589329. URL https://ieeexplore.ieee.org/document/9589329/.

Industrial Digital Twin Association - IDTA), 2023. *Specification of the Asset Administration Shell – Part 1: Metamodel (IDTA-01001-3-0, Version 3.0)*. PDF. URL https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2023/06/IDTA-01001-3-0_SpecificationAssetAdministrationShell_Part1_Metamodel.pdf.

Industrial Digital Twin Association - IDTA), 2024. *Specification of the Asset Administration Shell — Part 5: Package File Format (AASX)*. PDF. URL https://industrialdigitaltwin.org/wp-content/uploads/2024/06/IDTA-01005-3-0-1_SpecificationAssetAdministrationShell_Part5_AASXPackageFileFormat.pdf.

Isania Z. and Casalino G., 2025. *Discrete Event Simulation of a Stereolithography Production Line*. *Operations Research Forum*, 6, no. 4, 182. ISSN 2662-2556. doi:10.1007/s43069-025-00575-1. URL https://link.springer.com/10.1007/s43069-025-00575-1.

Kaya F.; Şanlı E.; Albayrak ; Ünal P.; and Kirci P., 2025. *Asset Administration Shell Tool Comparison: A Case Study with Real Digital Twins Used in Petrochemical Industry*. *Sensors*, 25, no. 7, 1978. ISSN

1424-8220. doi:10.3390/s25071978. URL https://www.mdpi.com/1424-8220/25/7/1978.

Krenczyk D.; Kempa W.M.; Kalinowski K.; Grabowik C.; and Paprocka I., 2018. *Integration of manufacturing operations management tools and discrete event simulation.* *IOP Conference Series: Materials Science and Engineering*, 400, 022037. ISSN 1757-899X. doi:10.1088/1757-899X/400/2/022037. URL https://iopscience.iop.org/article/10.1088/1757-899X/400/2/022037.

Lewicki W.; Niekurzak M.; and Wróbel J., 2024. *Development of a Simulation Model to Improve the Functioning of Production Processes Using the FlexSim Tool.* *Applied Sciences*, 14, no. 16, 6957. ISSN 2076-3417. doi:10.3390/app14166957. URL https://www.mdpi.com/2076-3417/14/16/6957.

Monek G.D. and Fischer S., 2023. *IIoT-Supported Manufacturing-Material-Flow Tracking in a DES-Based Digital-Twin Environment.* *Infrastructures*, 8, no. 4, 75. ISSN 2412-3811. doi:10.3390/infrastructures8040075. URL https://www.mdpi.com/2412-3811/8/4/75.

Pajpach M.; Šlauka M.; Pribiš R.; Drahoš P.; Kučera E.; and Haffner O., 2025. *Asset Administration Shell – Key-enabling technology of interoperability in Industry 4.0.* In *2025 Cybernetics & Informatics (K&I)*. ISSN 2767-875X, 1–6. doi:10.1109/KI64036.2025.10916454. URL https://ieeexplore.ieee.org/document/10916454/.

Sakurada L.; De La Prieta F.; and Leitao P., 2025. *Ten Years of Asset Administration Shell: Developments, Research Opportunities, and Adoption Challenges.* *IEEE Access*, 13, 127721–127741. ISSN 2169-3536. doi:10.1109/ACCESS.2025.3586716. URL https://ieeexplore.ieee.org/document/11072423/.

Santos R.; Piqueiro H.; Dias R.; and Rocha C.D., 2024. *Transitioning trends into action: A simulation-based Digital Twin architecture for enhanced strategic and operational decision-making.* *Computers & Industrial Engineering*, 198, 110616. ISSN 03608352. doi:10.1016/j.cie.2024.110616. URL https://linkinghub.elsevier.com/retrieve/pii/S0360835224007381.

Schutz J.; Sauvey C.; Nițu E.L.; and Gavriluță A.C., 2025. *A Practical and Sustainable Approach to Industrial Engineering Discrete-Event Simulation with Free Mathematical and Programming Software.* *Sustainability*, 17, no. 9, 3973. ISSN 2071-1050. doi:10.3390/su17093973. URL https://www.mdpi.com/2071-1050/17/9/3973.

Shi D.; Meyer O.; Fan Z.; Wang H.; and Bauernhansl T., 2026. *Cognitive digital twins for capability matching toward reconfigurable manufacturing: Leveraging asset administration shells and large language models.* *Robotics and Computer-Integrated Manufacturing*, 97, 103105. ISSN 07365845. doi:10.1016/j.rcim.2025.103105. URL https://linkinghub.elsevier.com/retrieve/pii/S0736584525001590.

Zhang J.; Ellwein C.; Heithoff M.; Michael J.; and Wortmann A., 2025. *Digital twin and the asset administration shell: An Analysis of the Three Types of AASs and their Feasibility for Digital Twin Engineering.* *Software and Systems Modeling*, 24, no. 3, 771–793. ISSN 1619-1366, 1619-1374. doi:10.1007/s10270-024-01255-0. URL https://link.springer.com/10.1007/s10270-024-01255-0.

Zhao J.; Aghezzaf E.H.; and Cottyn J., 2024a. *An extension of the Core Manufacturing Simulation Data standard to enhance the interoperability for discrete event simulation.* *Procedia CIRP*, 130, 1632–1637. ISSN 22128271. doi:10.1016/j.procir.2024.10.293. URL https://linkinghub.elsevier.com/retrieve/pii/S2212827124014501.

Zhao J.; Aghezzaf E.H.; and Cottyn J., 2024b. *Interoperability performance evaluation for discrete event simulation models: A step towards multilevel data exchange.* *Procedia CIRP*, 128, 72–77. ISSN 22128271. doi:10.1016/j.procir.2024.06.007. URL https://linkinghub.elsevier.com/retrieve/pii/S2212827124006516.