# Accelerating the Adoption of Asset Administration Shells through AI Agents

Camilo Velázquez-Rodríguez*, Steven Kauffmann*, Tom Pauwaert*, Stijn Huysentruyt†, Axl Van Alboom†
*Flanders Make, Leuven, Belgium; email: {camilo.velazquez, steven.kauffmann, tom.pauwaert}@flandersmake.be
†Universiteit Gent, Ghent, Belgium; email: {stijn.huysentruyt, axl.vanalboom}@ugent.be

***Abstract*—The adoption of Industry 4.0 practices has grown worldwide, driven by the need for standardised, transparent, and interoperable data exchange. The Asset Administration Shell (AAS) provides a standardised template for asset information, enabling companies to share and integrate data across systems. However, AAS creation from existing datasheets remains a manual and time-consuming process, hindering large-scale adoption. In this paper, we propose an Artificial Intelligence (AI) agent-based approach that automates the transformation of Portable Document Format (PDF) datasheets into AAS instances, which are then serialised into AAS files. The agents extract, structure, and validate asset information against the Industrial Digital Twin Association (IDTA) guidelines to ensure compliance with industry standards. We demonstrate the approach in a use case scenario, illustrating its potential to streamline the creation of AAS files and facilitate their adoption in a manufacturing environment.**

***Keywords*-Intelligent agents; Datasheet; AAS; Industry 4.0.**

## I. Introduction

The Industry 4.0 paradigm has revolutionised manufacturing processes by integrating advanced technologies, such as Internet of Things (IoT), Cyber-Physical Systems, and Artificial Intelligence [1]. This transformation has led to the emergence of smart factories. Nowadays, interconnected systems can communicate and make autonomous decisions to optimise production efficiency and flexibility [2]. A key enabler of this paradigm is the Asset Administration Shell (AAS), which serves as a digital representation of physical assets in the manufacturing environment [3].

The AAS provides multiple benefits. These include a standardised framework for managing and exchanging information about assets, facilitating interoperability among diverse systems and devices. By encapsulating all relevant data, functionalities, and services related to an asset, the AAS enables seamless integration into Industry 4.0 ecosystems. Additionally, the AAS supports advanced functionalities, such as predictive maintenance, real-time monitoring, and data analytics, which are crucial for optimising manufacturing processes [4]. Despite its potential, the adoption of AAS in industrial settings faces several challenges.

One of the main challenges is the complexity of the AAS implementation. New submodel templates released by the IDTA [5] and other standardisation bodies require significant effort to understand and apply correctly. In this regard, companies often struggle with the initial setup, configuration, and customisation of AAS for their specific use cases. Even though tools and frameworks exist to facilitate AAS creation, they often demand a steep learning curve and technical expertise. Therefore, manual processes, often chosen for AAS deployment, can be time-consuming and error-prone, hindering widespread adoption.

The lack of awareness among stakeholders is another barrier to AAS adoption. Many organisations are still unfamiliar with the benefits and functionalities of AAS, leading to resistance in embracing this relatively new technology. This is accompanied by the need for skilled personnel to manage and maintain AAS-based systems [6]. Many of the companies, although interested in Industry 4.0, lack the necessary expertise to effectively implement submodel templates and further utilise AAS in their operations. For example, Small and Medium-sized Enterprises (SMEs) may find it particularly challenging to allocate resources for training and development in this area [7]. In addition, many industries still have plenty of machinery and equipment not Industry 4.0-ready, making the integration of AAS even more complex [8].

To address some of the mentioned challenges, there is a growing interest in leveraging automated solutions to streamline the adoption process. In this context, Artificial Intelligence (AI) agents have emerged as promising tools to facilitate the deployment and management of AAS in manufacturing environments. AI agents can autonomously perform tasks, such as data collection, analysis, and decision-making, thereby reducing the burden on human operators [9]. By integrating AI agents with AAS, it is possible to enhance the efficiency and effectiveness of asset management processes, ultimately accelerating the adoption of AAS in Industry 4.0 settings.

This paper explores the potential of AI agents to support the adoption of AAS in Industry 4.0 environments. More specifically, we investigate how AI agents can assist in the automatic creation of AAS instances from PDF datasheets. Generated AAS instances can then be later integrated into a component library to speed up the commissioning of physical assets. PDF datasheets are commonly used to document technical specifications of assets. Oftentimes, they contain valuable information that can be utilised to populate AAS submodel templates. However, manually extracting and structuring this information can be labour-intensive and prone to errors. Our paper makes the following contributions:

- Automatic generation of AAS from PDF datasheets using AI agents and knowledge graphs.
- A case study demonstrating the application of our approach in generating AAS instances for an industrial robot arm, highlighting the practical implications and benefits.
- The discussion of challenges faced during implementation and future directions in this area.

While our approach shows promise, certain limitations are acknowledged: generated AAS quality depends on input data completeness, manual intervention remains necessary for eClass classification, and organisations with heavily customised legacy systems may face adoption barriers.

The remainder of the paper is organised as follows: In Section II, we review the related work on AAS adoption and AI agents in Industry 4.0. Section III provides background information on AAS, AI agents, knowledge graphs, etc. Section IV presents our approach for automatic AAS generation based on the automated information extraction from PDF datasheets. Section V describes the case study, experimental setup, and faced challenges during implementation. Section VI discusses the results, their implications, and potential limitations of our approach. Finally, Section VII concludes the paper and outlines future research directions.

## II. Related Work

### A. Traditional Agents for AAS Adoption

The interaction of agents with assets in industry has been broadly studied over the past few years. Precisely, the definition of Multi-Agent Systems (MAS) by Wooldridge [10] laid the foundation for agents in industry. As per [10], a MAS is a "*society of intelligent, cooperative, proactive, and autonomous entities*" (i.e., agents). These agents rely on symbolic reasoning, explicit knowledge representations (e.g., logic-based systems), and planning algorithms to make decisions. They are designed for specific tasks, such as negotiation, coordination, or resource allocation in MAS. Their intelligence is rule-based or logic-based, with limited adaptability outside predefined domains. Communication is structured, often relying on predefined protocols (e.g., Foundation for Intelligent Physical Agents - Agent Communication Language (FIPA-ACL)) and ontologies for inter-agent interactions. Therefore, we will refer to these as *traditional agents* for the remainder of the paper.

Based on the previous MAS definition, a group of works applied it to facilitate the adoption of AAS in industry [6, 11–17]. They leverage MAS for the implementation, digitisation, and adoption of AAS. These works describe the use of different traditional agents in conjunction with AAS to enhance standardisation and communication between physical assets and their digital twin counterparts. Many assume that AAS files are already filled with the required information in a specific submodel and leverage their approach on this assumption. In reality, the instantiation of an AAS submodel might take several hours (depending on the asset and submodel complexities), making it a labour intensive process.

Our approach makes use of Large Language Model (LLM)-based models, in contrast to traditional agents, such as in the previous works. Modern agents interact with a transformer-based neural network trained on massive datasets, also known as an LLM. This can be general-purpose or fine-tuned for a specific task. LLM-based agents can operate autonomously but often rely on human prompts or predefined workflows to initiate actions. These agents excel in natural language communication, interacting through conversational interfaces without requiring formal protocols. Unlike Wooldridge's agents [10], LLM-based agents demonstrate emergent general-purpose reasoning, enabling them to handle a wide range of tasks without domain-specific programming.

### B. LLMs (Agents) for AAS Adoption

The use of LLMs to facilitate the adoption of AAS in industry is a novel approach. Prompts, as well as agents powered by LLMs, have been used to generate AAS files from unstructured data [18]. Xia et al., [18] propose a framework that leverages LLMs to interpret text-based data and generate AAS-compliant files. They based their approach on the capabilities of LLMs to match semantically similar concepts of syntactically different terminologies. This can be useful when dealing with unstructured data from various sources, as LLMs can understand context and meaning beyond rigid schemas.

Although we also use LLMs to syntactically map different terminologies, our approach goes beyond that of Xia et al., [18] in several ways. First, our approach automatically extracts text information from PDF datasheets, instead of relying on text-based data as input. Second, our process involves a knowledge database creation and a querying step to prevent hallucinations from the LLM. Third, we also map eClass IDs to the extracted data to ensure that the generated AAS files are compliant with industry standards. We are not aware of any other work that uses LLMs to facilitate the adoption of AAS in industry apart from the work of Xia et al., [18].

## III. Background

### A. AI Approaches and Techniques

*1) Vector databases:* Vector databases are specialised systems for storing and indexing high-dimensional embeddings, enabling efficient approximate similarity search (e.g., nearest neighbour) over data, such as text, images or audio [19].

*2) Large Language Models:* LLMs are transformer-based neural networks pre-trained on massive corpora, which scale in parameter count, data, and compute, exhibiting emergent capabilities without task-specific tuning. While their applications span across many domains, main limitations include data bias, hallucination, high inference/training costs, and challenges in evaluation and governance [20, 21].

*3) AI Agents:* AI agents are systems that not only generate responses but perceive, plan, decide and act (often using tools, memory, and environment interactions) to achieve goals with some degree of autonomy [22].

*4) Knowledge graphs:* Knowledge graphs are directed, labelled, multi-relational graphs in which nodes represent entities and edges represent semantic relationships. These graphs are often enriched with an ontology and are used to integrate, query, and reason over structured and semi-structured data. They provide explainability and support complex relational queries and inference. Some of their challenges include scalability, schema alignment, entity/edge extraction, and maintaining/updating large, heterogeneous graphs [23–25].

### B. Asset Administration Shell

The Asset Administration Shell (AAS) is a standardised digital representation of industrial assets throughout their lifecycle. An AAS file can contain one or more AAS instances, where each instance represents a digital twin of a specific physical or logical asset. AAS files are organised via modular submodels, which capture technical, operational, semantic, and communication aspects enabling interoperability across heterogeneous systems and stakeholders [17, 18, 26–29]. The Industrial Digital Twin Association (IDTA) [30] publishes, maintains and distributes AAS submodel template files publicly. Currently, there are various submodels available for specific target domains (e.g., digital nameplate for industrial equipment, functional safety, technical data, etc.) [5], while others are intended to be released in the near future (e.g., carbon footprint, maintenance instructions, etc.).

Each AAS submodel defines specific fields to be filled in, which correspond to the characteristics of the asset being digitalised. Fields also have types (e.g., date, integer, string) depending on the data to be stored. An asset might have more characteristics than those specified by the submodel template. To handle these differences, some submodel templates (e.g., digital nameplate) can be extended with custom fields. These fields are of string type to be completed with generic information, and not specific to the submodel template.

## IV. Approach

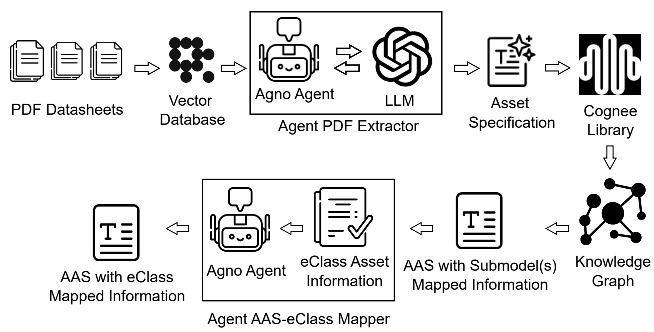


Figure 1. Approach to automatically move from PDF datasheets to AAS files.

Figure 1 shows the implemented approach. Datasheets in the form of PDF files represent the input to the approach. These are distributed as part of the documentation for different assets (e.g., robot arm, industrial pump, etc.). Additionally, they describe the main asset characteristics to be extracted later on.

A vector database is used to process the documents by extracting text and assigning a vector to each extracted token. Extracted words and their corresponding vectors will later be used to map characteristics that might be syntactically different but semantically similar. Currently, multiple vector database tools exist providing help to automatic information extraction and vector matching. We selected LanceDB [31] due to its integration with the agent framework we use. Our approach is not limited to this specific vector database, and other tools can be used as well.

Once the documents are stored, processed and vectorised, an AI agent will interact with this information and a pre-established Large Language Model (LLM). The selection of LLMs also depends on many factors related to the task at hand. For example, some models are better suited for code generation, while others excel at natural language understanding. We selected the OpenAI models [32] to interact with the agent, as they have demonstrated state-of-the-art performances for many different tasks. Specifically, we used the *Generative Pre-trained Transformer (GPT)-4.1* model to extract assets' characteristics from the documents.

The agent framework Agno [33] provides the necessary tools for our approach. Agno's integration with the LanceDB vector database and other services and models (e.g., OpenAI) allowed us to focus on the agents' configuration, instead of implementation details. Previously processed datasheet documents serve as a knowledge base, whilst *GPT-4.1* is used as a language model. We also make use of the internal reasoning tool of the Agno framework in the agent configuration. This tool instructs the agent to solve the task at hand as a series of logical steps and to trace back the reasoning process. Reasoning includes analysis and revision of the subtasks' performance. The configured prompt instructs the agent to extract technical information about one specific asset at a time.

The result of running the agent, given the previous configuration specifications, is a file containing extracted technical information about the specified asset. As the information of different assets may be heterogeneous and diverse, the language model might tend to hallucinate. This phenomenon has been identified as a drawback for these models [34, 35]. Therefore, we opted to address this issue by relying solely on the extracted information from the datasheets. We assume that the datasheets contain the necessary information to populate the AAS templates.

The Cognee library [36] provides an approach to solve the hallucination related to LLMs. The library enforces agents to rely on the information from a knowledge graph which is created from a set of pre-defined sources (e.g., PDF datasheets). This knowledge graph is generated in the form of a Neo4J database [37] where nodes represent automatically extracted entities and edges their corresponding relationships. Once the knowledge graph is created (cf., Figure 1), responses to queries are limited to only the scope of the information within the graph. Thus, a hallucination is not possible by construction, as the agent cannot invent information not present in the graph.

Figure 1 also shows that after the Neo4J knowledge graph creation, queries are requested to the database. Cognee enables natural language queries, which are transformed into the Cypher graph query language. Our approach further leverages this feature by mapping extracted asset characteristics to the fields in an AAS template file. Asset characteristics and AAS fields may differ syntactically while conveying similar semantics.

An instantiation of the specified AAS templates is the outcome of the previous mapping step. The AAS template files are based on the IDTA guidelines. More specifically, in this work, we used the AAS template for Digital Nameplate, Technical Data, and Functional Safety. The three instantiations are then combined into one AAS file per asset. These can be found in the IDTA AAS Template Repository on GitHub [38].

The inclusion of additional templates is part of future work.

A final mapping step is performed to enrich the AAS instantiation with additional information. eClass [39] IDs are standard identifiers for asset characteristics that might not be part of the default information in templates (i.e., custom-added fields). Therefore, we mapped such custom-added characteristics to their corresponding eClass IDs. For this purpose, we used another AI agent configured similarly to the previous one.

The outcome of our approach is an AAS file per asset, which is automatically generated from the corresponding datasheets. These AAS files can then be used in Industry 4.0 applications to represent assets in a standardised manner. They contain the main technical information as well as additional eClass IDs for custom-added fields.

## V. Use Case

We present a use case that illustrates the practical application of our proposed method. The use case focuses on a real-world scenario where PDF datasheets are processed by our approach to generate AAS files and be used in an Industry 4.0 context. We describe the specific steps taken in this use case, the encountered challenges, and the achieved results.

### A. Scenario Description and Resolution

*1) Description:* The use case involves a manufacturing company that aims to digitise its asset information by converting existing PDF datasheets into AAS files. The company has a large repository of datasheets for various components and equipment used in its production processes. We will focus on an asset type to demonstrate the effectiveness of our approach.

We focus our approach on robotic arms used in assembly lines. We consider a 6-axis articulated robotic arm commonly used for pick-and-place operations. More specifically, we use the datasheets provided by the manufacturer Stäubli for their TX2-90XL model [40]. This robotic arm is equipped with various sensors and actuators, and its datasheets contain detailed technical specifications, performance data, and operational guidelines. The datasheets provide information for three variants of the TX2-90 model (e.g., the TX2-90, TX2-90L, and TX2-90XL), each with different payload capacities and reach. Precisely, we selected these datasheets as they have a complex structure, including multiple tables, images, and technical diagrams, which pose challenges for automated data extraction.

Characteristics

| | TX2-90 | TX2-90L | TX2-90XL |
|---|---|---|---|
| Load capacity | 14 kg | 12 kg | 7 kg |
| Reach (between axis 1 and 6) | 1000 mm | 1200 mm | 1450 mm |
| Number of degrees of freedom | 6 | 6 | 6 |
| Repeatability - ISO 9283 | ± 0.02 mm | ± 0.02 mm | ± 0.02 mm |
| Weight | 114 kg | 117 kg | 119 kg |
| UL certification | ✓ | ✓ | ✓ |

Figure 2. Screenshot of the Stäubli TX2-90XL robotic arm datasheet.

Figure 2 shows a table extracted from the datasheet, which lists the technical specifications of the TX2-90 robotic arm variants. Since this use case focuses on the robot arm TX2-90XL, our approach needs to correctly identify and extract the relevant information from the table in Figure 2. Additionally, other relevant information that might be in multiple documents also needs to be extracted. This can include the robot's degrees of freedom, maximum speed, repeatability, and operating conditions.

*2) Resolution:* As previously described in Section IV, our approach involves several steps to generate AAS files from PDF datasheets. First, the PDF datasheets are processed using a vector database to extract and vectorize the textual information. Next, a smart agent configured with the Agno framework interacts with the vector database and a pre-established LLM (GPT-4.1) to extract the relevant technical specifications of the TX2-90XL robotic arm. The agent is prompted to focus on the specific asset and extract information, such as payload capacity, reach, degrees of freedom, maximum speed, repeatability, and operating conditions. A manual review of the extracted information is performed to ensure accuracy and completeness. The extracted information is then mapped to the corresponding fields in an AAS template file using the Cognee library. A second mapping step is performed to link the extracted information to the appropriate eClass IDs. Such a mapping is performed using the eClass database to ensure that each field in the AAS file is correctly identified and classified. The mapped information is written into the AAS templates, resulting in a complete AAS file for the TX2-90XL robotic arm. Finally, the generated AAS file is validated against the AAS schema to ensure compliance with Industry 4.0 standards.

### B. Challenges

During the execution of the use case, several challenges were encountered. One of the main challenges was dealing with the heterogeneity of the datasheet formats. The datasheets contained various layouts, tables, and images, which made it difficult for the agent to extract the relevant information. To address this challenge, we leveraged the capabilities of the vector database to effectively index and retrieve the textual information. Additionally, the heterogeneity of the language used in the datasheets posed a challenge in case this process would be performed manually. Therefore, we relied on the LLM's ability to understand and process technical language and similar terminologies.

Each of the mappings (to the AAS template and to the eClass IDs) also presented challenges. The AAS template needed to be analysed to accommodate the specific attributes of the TX2-90XL robotic arm. The mapping to eClass IDs required access to an up-to-date eClass database and understanding of the classification system. To overcome these challenges, we ensured that the AAS template was flexible enough to allow for assets that may not appear in the standard. Also, we manually focus on the robotic arm domain to ensure that the correct eClass IDs were assigned. In this regard, future work could explore automating the eClass mapping process further.

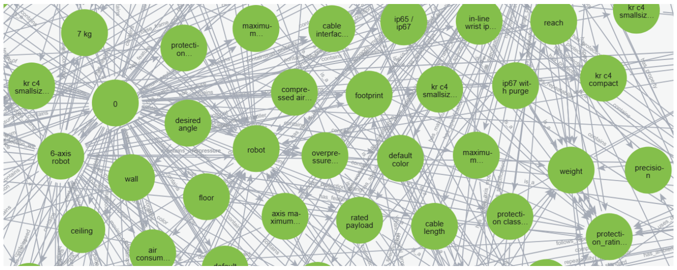


Figure 3. Screenshot of the Neo4J knowledge graph created from the available datasheets.

Lastly, another challenge was ensuring the accuracy of the extracted information. As we relied on official datasheets, and also based on a knowledge base with the extracted information, we mitigated the risk of inaccuracies. Despite this, we still performed a manual review of the extracted information to ensure its correctness.

### C. Results

The results of the use case proved the effectiveness of our approach in generating AAS files from PDF datasheets. One of the first results we obtained was the knowledge graph in the Neo4J database. This graph served as a valuable resource for querying and retrieving information about the robotic arm.

Figure 3 shows a screenshot of the knowledge graph in Neo4J, illustrating the relationships between different technical specifications of the TX2-90XL robotic arm. Queries can be in natural language which are then translated into Cypher queries to retrieve specific information. For example, a query such as *"What is the maximum payload of the TX2-90XL robotic arm?"* would return this specific information from the graph.

The generated AAS instance for the TX2-90XL robotic arm contained the technical specifications extracted from the datasheets. Figure 4 shows a screenshot of the generated AAS instance in the visualisation tool AASX Package Explorer [41].

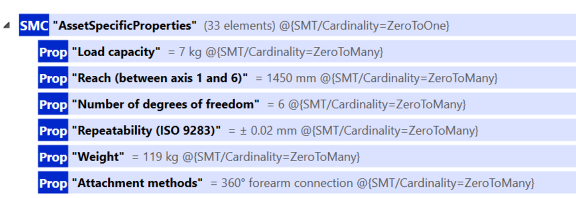


Figure 4. Screenshot of the generated AAS instance for the Stäubli TX2-90XL robotic arm in the AASX Package Explorer.

Our approach also includes the datasheet PDFs as part of the AAS file, ensuring that the original documentation is available for reference. A visualisation tool supporting PDF rendering can display the datasheets directly from the generated file.

## VI. Discussion

### A. Implications

The proposed approach has several implications for the adoption of AAS in industrial settings. By leveraging AI agents to automate the creation of AAS instances and files, we can reduce the time and effort required for their implementation. This can lead to faster integration of AAS into existing systems, promoting interoperability and standardisation across different platforms. Previously considered tedious and error-prone tasks can now be streamlined, allowing organisations to focus on leveraging the benefits of AAS.

The integration of industries into the Industry 4.0 paradigm could be further accelerated by this approach. With the ability to quickly generate AAS instances and files, industries can more readily adopt digital twins and other Industry 4.0 technologies, leading to improved efficiency in manufacturing processes. The use of AI agents also opens up new possibilities for customisation and scalability, as AAS can be tailored to specific use cases and easily adapted to changing requirements.

### B. Limitations

Despite the promising results, there are several limitations to our approach that need to be addressed in future work. First, the quality of the generated AAS instances is heavily dependent on the input data and the capabilities of the AI agents. Inaccurate or incomplete data can lead to suboptimal AAS instances, which may not fully meet the requirements of the intended application. Second, the current implementation may not fully capture the complexity of certain industrial scenarios. While AI agents are capable of generating AAS files for a wide range of use cases, there may exist specific cases that require more sophisticated modelling and representation. Finally, there is still a need for manual intervention in the eClass classification process, which may introduce delays.

Beyond mentioned technical limitations, practical challenges exist regarding adoption and use by existing organisations. Many industrial environments have heavily customised legacy systems and established workflows deeply integrated into their operations. Introducing AAS-based approaches may require significant organisational change management. Furthermore, organisations with investments in proprietary data management systems may face resistance to adopting AAS formats, particularly if migration costs are perceived as prohibitive.

## VII. Conclusion and Future Work

In this paper, we presented an approach leveraging AI agents to facilitate AAS creation from PDF datasheets in industrial settings. We utilize the Cognee library to create a knowledge graph from datasheet information, which AI agents use to generate and populate the AAS structure. A use case involving an industrial robot arm demonstrates the potential of this approach to automate the AAS creation process. Future work will focus on extensive evaluations in real-world industrial environments and integration of generated AAS files into component libraries for flexible manufacturing systems. This enables dynamic asset assignment, reducing manual programming effort and increasing production flexibility.

## References

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld and M. Hoffmann, ‘Industry 4.0’, *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] Y. Lu, ‘Industry 4.0: A survey on technologies, applications and open research issues’, *Journal of industrial information integration*, vol. 6, pp. 1–10, 2017.

[3] E. Tantik and R. Anderl, ‘Integrated data model and structure for the Asset Administration Shell in Industrie 4.0’, *Procedia Cirp*, vol. 60, pp. 86–91, 2017.

[4] C. Wagner et al., ‘The role of the Industry 4.0 Asset Administration Shell and the Digital Twin during the life cycle of a plant’, in *2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA)*, IEEE, 2017, pp. 1–8.

[5] IDTA, ‘IDTA Submodel Templates’, Accessed: 2025-01-06, 2024. [Online]. Available: https://industrialdigitaltwin.org/en/content-hub/submodels.

[6] A. Sidorenko et al., ‘The MAS4AI framework for human-centered agile and smart manufacturing’, *Frontiers in Artificial Intelligence*, vol. 6, p. 1 241 522, 2023.

[7] F. Faiz, V. Le and E. K. Masli, ‘Determinants of digital technology adoption in innovative SMEs’, *Journal of Innovation & Knowledge*, vol. 9, no. 4, p. 100 610, 2024.

[8] A. Alqoud, D. Schaefer and J. Milisavljevic-Syed, ‘Industry 4.0: a systematic review of legacy manufacturing system digital retrofitting’, 2022.

[9] B. Wang, X. Li, T. Freiheit and B. I. Epureanu, ‘Learning and intelligence in human-cyber-physical systems: Framework and perspective’, in *2020 Second International Conference on Transdisciplinary AI (TransAI)*, IEEE, 2020, pp. 142–145.

[10] M. Wooldridge, *An Introduction to Multiagent Systems*. John wiley & sons, 2009.

[11] L. Sakurada and P. Leitão, ‘Multi-Agent Systems to Implement Industry 4.0 Components’, in *2020 IEEE conference on industrial cyberphysical systems (ICPS)*, IEEE, vol. 1, 2020, pp. 21–26.

[12] A. López, O. Casquero and M. Marcos, ‘Design patterns for the implementation of Industrial Agent-based AASs’, in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, IEEE, 2021, pp. 213–218.

[13] B. Vogel-Heuser, F. Ocker and T. Scheuer, ‘An approach for leveraging Digital Twins in agent-based production systems’, *at-Automatisierungstechnik*, vol. 69, no. 12, pp. 1026–1039, 2021.

[14] S. Jungbluth et al., ‘Dynamic Replanning using Multi-Agent Systems and Asset Administration Shells’, in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2022, pp. 1–8.

[15] L. Sakurada, P. Leitao and F. De La Prieta, ‘Engineering a Multi-agent Systems Approach for Realizing Collaborative Asset Administration Shells’, in *2022 IEEE International Conference on Industrial Technology (ICIT)*, IEEE, 2022, pp. 1–6.

[16] L. Sakurada, P. Leitao and F. De la Prieta, ‘Agent-Based Asset Administration Shell Approach for Digitizing Industrial Assets’, *Ifac-Papersonline*, vol. 55, no. 2, pp. 193–198, 2022.

[17] L. Sakurada, F. De La Prieta and P. Leitao, ‘Ten Years of Asset Administration Shell: Developments, Research Opportunities, and Adoption Challenges’, *IEEE Access*, 2025.

[18] Y. Xia, Z. Xiao, N. Jazdi and M. Weyrich, ‘Generation of Asset Administration Shell with Large Language Model Agents: Toward Semantic Interoperability in Digital Twins in the Context of Industry 4.0’, *IEEE Access*, vol. 12, pp. 84 863–84 877, 2024.

[19] Y. Han, C. Liu and P. Wang, ‘A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge’, *arXiv preprint arXiv:2310.11703*, 2023.

[20] W. X. Zhao et al., ‘A Survey of Large Language Models’, *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[21] K. S. Kalyan, ‘A survey of GPT-3 family large language models including ChatGPT and GPT-4’, *Natural Language Processing Journal*, vol. 6, p. 100 048, 2024.

[22] P. Zhao, Z. Jin and N. Cheng, ‘An In-depth Survey of Large Language Model-based Artificial Intelligence Agents’, *arXiv preprint arXiv:2309.14365*, 2023.

[23] A. Hogan et al., ‘Knowledge Graphs’, *ACM Computing Surveys (Csur)*, vol. 54, no. 4, pp. 1–37, 2021.

[24] S. Ji, S. Pan, E. Cambria, P. Marttinen and P. S. Yu, ‘A Survey on Knowledge Graphs: Representation, Acquisition, and Applications’, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2021.

[25] C. Peng, F. Xia, M. Naseriparsa and F. Osborne, ‘Knowledge Graphs: Opportunities and Challenges’, *Artificial intelligence review*, vol. 56, no. 11, pp. 13 071–13 102, 2023.

[26] S. Rongen, N. Nikolova and M. van der Pas, ‘Modelling with AAS and RDF in Industry 4.0’, *Computers in Industry*, vol. 148, p. 103 910, 2023.

[27] J. Arm et al., ‘Automated Design and Integration of Asset Administration Shells in Components of Industry 4.0’, *Sensors*, vol. 21, no. 6, p. 2004, 2021.

[28] L. Sakurada, F. De la Prieta and P. Leitao, ‘The Role of Multi-Agent Systems in Realizing Asset Administration Shell Type 3’, *Future Internet*, vol. 17, no. 7, p. 270, 2025.

[29] A. Alexopoulos, G. Kalogeras, K. Koutras and A. Kalogeras, ‘Why Asset Administration Shells: A Survey on Uses and Challenges’, *IEEE Access*, 2025.

[30] IDTA, ‘Industrial Digital Twin Association’, Accessed: 2025-01-06, 2024. [Online]. Available: https://industrialdigitaltwin.org/en/.

[31] LanceDB, ‘LanceDB: Developer-friendly, serverless vector database’, Accessed: 2025-01-06, 2024. [Online]. Available: https://lancedb.com/.

[32] OpenAI, ‘OpenAI Platform Models’, Accessed: 2025-01-06, 2024. [Online]. Available: https://platform.openai.com/docs/models.

[33] Agno, ‘Agno: AI Agent Framework’, Accessed: 2025-01-06, 2024. [Online]. Available: https://docs.agno.com/introduction.

[34] Z. Ji et al., ‘Towards Mitigating Hallucination in Large Language Models via Self-Reflection’, in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 1827–1843.

[35] A. Martino, M. Iannelli and C. Truong, ‘Knowledge Injection to Counter Large Language Model (LLM) Hallucination’, in *European Semantic Web Conference*, Springer, 2023, pp. 182–185.

[36] Cognee, ‘Cognee: Knowledge Graph Generation’, Accessed: 2025-01-06, 2024. [Online]. Available: https://www.cognee.ai/.

[37] Neo4j, Inc., ‘Neo4j Graph Database’, Accessed: 2025-01-06, 2024. [Online]. Available: https://neo4j.com/.

[38] IDTA, ‘IDTA AAS Submodel Templates Repository’, Accessed: 2025-01-06, 2024. [Online]. Available: https://github.com/admin-shell-io/submodel-templates.

[39] eCl@ss e.V., ‘eCl@ss: International Standard for the Classification and Description of Products and Services’, Accessed: 2025-01-06, 2024. [Online]. Available: https://eclass.eu/en/.

[40] Stäubli Robotics, ‘Stäubli TX2-90XL Industrial Robot’, Accessed: 2025-01-06, 2024. [Online]. Available: https://www.staubli.com/us/en/robotics/products/industrial-robots/tx2-90.html.

[41] Eclipse AASPE, ‘AASX Package Explorer’, Accessed: 2025-01-06, 2024. [Online]. Available: https://github.com/eclipse-aaspe/package-explorer.