

From Automation to Autonomy: A Digital Twin Framework for transparent Agent and Human Collaboration in Industrial Multi-Agent Systems

Inga Miadowicz ^{1*} , Mathias Kuhl ¹, Daniel Maldonado Quinto ¹, Robert Pitz-Paal ¹, and Michael Felderer ¹

¹ Deutsches Zentrum für Luft- und Raumfahrt (DLR); Inga.Miadowicz@dlr.de, Mathias.Kuhl@dlr.de, Daniel.MaldonadoQuinto@dlr.de, Robert.Pitz-Paal@dlr.de, Michael.Felderer@dlr.de

* Correspondence: Inga.Miadowicz@dlr.de

Abstract

With the advancement of digitization in the era of Industry 4.0 (I4.0), highly automated, semi-autonomous, and fully autonomous systems are emerging. Within this context, multi-agent systems (MAS) offer a promising approach for automating tasks and processes based on autonomous agents that work together in an overall system to increase the degree of system autonomy stepwise in a modular and flexible way. A critical research challenge is determining how these agents can collaboratively engage with both other agents and human operators to facilitate the gradual transition from automated to fully autonomous industrial systems. To close transparency and connectivity gaps, this study contributes with a framework for the collaboration of agents and humans in increasingly autonomous MAS based on a Digital Twin (DT). The framework specifies a standard-based data model for MAS representation and proposes to introduce a DT infrastructure as a service layer for system coordination, supervision, and interaction. To demonstrate the feasibility and assess the quality of the framework, it is implemented and evaluated in a case study in a real-world industrial scenario. As a result of the study, we infer that the DT framework offers significant benefits in facilitating transparent and seamless cooperation between agents and humans within increasingly autonomous industrial MAS.

Keywords: Autonomous System, Multi-Agent System, DT, Industry 4.0, Collaboration, Self-X

Received:

Revised:

Accepted:

Published:

Citation: Miadowicz, I.; Maldonado Quinto, D.; Pitz-Paal, R.; Felderer, M. From Automation to Autonomy: A Digital Twin Framework for transparent Agent and Human Collaboration in Industrial Multi-Agent Systems. *Systems* **2025**, *1*, 0. <https://doi.org/>

Copyright: © 2025 by the authors. Submitted to *Systems* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In today's rapidly evolving industrial landscape, there is an increasing demand for more efficient, flexible, robust, and cost-effective production systems. Technological advances from Industry 4.0 (I4.0) are enabling a transformative shift that evolves industrial system infrastructures to highly adaptable and autonomous systems [1]. These advanced systems are expected to execute complex tasks and processes in a data-driven and intelligent manner, with minimal human intervention, while meeting the high demands of modern industrial production [2].

As industry strives to meet these high expectations, agent-based approaches have emerged as a promising solution. Thus, industrial processes are stepwise automated in a modular way, based on the introduction of intelligent agents that take over responsibility for individual tasks [3,4]. However, despite the progress made in automating single process tasks, significant challenges remain when it comes to the collaboration of autonomous

agents in multi-agent systems (MASs) for a flexible, efficient, and safe end-to-end automation of entire industrial processes [5].

In particular, the gradual transition from classical automation systems to autonomous systems, in which agents gradually automate tasks and work alongside human operators, raises complex questions about system coordination, supervision, and interaction [6]. Traditional industrial automation systems typically allow operators to monitor and control industrial processes from a centralized control room through the use of Human-Machine Interfaces (HMIs) [7]. Thus, these systems differ substantially from MASs, which execute industrial processes through a decentralized network of agents. While agents cooperate internally using, for example, custom software interfaces or a specialized agent communication language [8,9], they typically function as black boxes, lacking unified interfaces for monitoring their internal state, invoking their services, and understanding their role in the overall structure of the MAS [6,7]. The lack of transparency and unified interfaces for cooperation with other agents and human operators introduces critical gaps in the evolutionary development and supervision of increasingly autonomous MAS. To bridge these gaps, agents must be able to communicate their roles, capabilities, and operational states within the system. In addition, they should offer interfaces that enable collaboration with both other agents and human actors for seamless automation of processes. This transparency and connectivity are essential for the trusted, flexible, and successful gradual transition toward agent-based adaptive and autonomous systems in industrial environments.

This article aims to address these challenges by exploring the potential of a Digital Twin (DT) to improve transparency and support inter-agent and human-agent collaboration in industrial MASs. Thereby, this study proposes a framework using DTs not only as a representation of the physical space but also to virtually represent agents in the MAS itself. On one hand, the DT serves as a shared medium for agents to represent themselves, monitor, and discover other agents in the system, and request task execution for the collaboration of agents to enable the seamless automation of processes within the MAS. On the other hand, the data-driven DT approach is used to implement interfaces to humans, enabling human operators to transparently supervise agent-based process execution. Through this approach, we strive to enable smoother transitions toward fully autonomous systems while ensuring robust monitoring, control, and error handling during the gradual automation process.

The research question driving this study is: *How can the integration of DTs as a service infrastructure for increasingly autonomous MASs overcome challenges related to transparency and inter-agent and human-agent collaboration in the industrial domain?* To answer the research question, this study first gives an overview of the theoretical background of autonomous systems, MAS, and DTs. Second, the current state-of-the-art with respect to DTs as a baseline for industrial MAS is summarized, and the novelty of the work is emphasized. Third, the research method is presented to demonstrate how the study systematically generates new knowledge through a design-centered approach. Subsequently, the framework for agent collaboration based on a DT is introduced, and the proposed concepts are demonstrated and evaluated in a case study to emphasize how the framework can be applied in a real-world industrial scenario and how suitable it is for its intended purpose. Finally, the discussion and conclusion sections reflect the results of the study and explicitly answer the underlying research question.

2. Background

First, an overview of the fundamental theoretical background of this work is given, which encompasses autonomous systems, MASs, and DTs.

2.1. Autonomous Systems

An autonomous system is commonly understood in state-of-the-art research as a system capable of achieving its objectives within uncertain environments with minimal or no external intervention [2,10–12]. Because there is currently no universally accepted standard for what exactly constitutes an autonomous system [13], the precise definition of autonomous systems remains uncertain. The result is that system autonomy is characterized differently in various application domains [14,15].

The term ‘autonomous’ originally translates into self-governance, which in a technical context implies independent decision-making, adaptation, and action with little or no human involvement [2,16]. Autonomous industrial systems typically possess the ability to sense, analyze and interact with their environment, enabling them to pursue objectives despite unpredictability or change. This self-sufficiency distinguishes them from automated systems, which follow predefined rules in stable contexts and lack robust adaptation to new or uncertain situations [17].

System autonomy is widely acknowledged as a spectrum, not a binary state, with systems evolving through incremental advancements toward greater degrees of autonomy [1,12]. The trend is to develop systems with increasingly advanced self-x capabilities such as self-learning, self-management, or self-adaptation. This ongoing process can be seen as a ‘quest for autonomy’ that has preoccupied humanity for decades [18]. The development of industrial autonomous systems is still in an early stage compared with research efforts in other domains such as space or military. Although there are already some authors who discuss first concepts of system autonomy in the industrial field [1,2,13,17,18], the level of automation in the industrial domain generally remains low compared to other sectors [1].

In summary, the concept of system autonomy lacks standardization, reflecting diverse interpretations in the current state of research concerning the criteria that characterize autonomous systems. Generally, autonomous systems are defined by their independent pursuit of goals in uncertain environments with limited human intervention. Thereby, systems usually evolve towards higher degrees of autonomy on the basis of ongoing technological progress.

2.2. Multi-Agent Systems

A MAS consists of multiple interacting agents that collectively work to solve complex tasks that are difficult or impossible for a single agent to handle. Each agent is responsible for accomplishing one or more tasks and collaborates with other agents to achieve global system goals in a distributed and decentralized manner [9].

Accordingly, an agent is the fundamental unit of a MAS. It is defined as an entity situated in an environment that senses various parameters and, based on these perceptions in conjunction with its goals, makes decisions and performs actions to achieve those goals. Such an entity may be a software component, hardware component, or a hybrid of both. The environment refers to the context or location in which the agent operates, such as a network or a software platform [8].

MAS have emerged as a transformative technology in the industrial sector, effectively addressing challenges such as rapidly evolving market conditions, globally distributed production networks, and the increasing complexity of industrial systems [3,4]. By decomposing complex industrial systems into smaller subsystems - each implemented as decentralized, cooperative, and intelligent agent [3,4] - MAS improve flexibility, autonomy, and operational efficiency [5]. Consequently, MAS technology has been applied to automate a wide range of industrial use cases, including process control, production planning, scheduling, monitoring, quality control, or fault diagnosis [5].

Agent-based strategies are recognized for their potential benefits, but practical deployment in industrial environments is still relatively rare [4,5,19]. Key difficulties in MAS involve the coordination of collaboration between autonomous agents to ensure that local actions culminate in the fulfillment of global objectives in a reliable way [8,9]. Furthermore, there are research gaps related to the cooperation of agents with humans lacking user interfaces for monitoring and control [6,7] and missing real-world applications that demonstrate and evaluate the application of agent-based approaches in industrial environments [20].

In summary, a MAS is a network of agents that collaborate to achieve the overall goals of the system. Challenges in the industrial domain include, but are not limited to, inter-agent collaboration, human-agent cooperation, and studies in real-world system environments. Closing research gaps related to transparency and connectivity is essential to pave the way for a successful and trusted agent-based gradual transition toward autonomous systems and their widespread adoption in the industrial domain.

2.3. DTs

A DT is essentially a digital representation or virtual replica of a physical asset, such as a product, machine, or entire system. This virtual model reflects the real-world object and allows for simulation, analysis, and monitoring throughout its lifecycle [21]. Due to its high potential in different application domains such as industry, health or smart cities, the interest in DTs has grown considerably in recent years [22].

The terms Digital Model (DM), Digital Shadow (DS), and DT, although related, describe different levels of digital representation and interaction between the physical and digital worlds. An authoritative classification by Kritzinger et al. [23] differentiates them according to the degree and direction of the data flow as illustrated in Figure 1. Thereby, a DM is a static representation with no automatic synchronization of data between the virtual representation with its physical counterpart. A DS automatically receives data from the physical environment and mirrors the latest up-to-date information of physical systems or components. Finally, the DT is a bi-directional representation where changes in the physical asset update the virtual twin and vice versa.

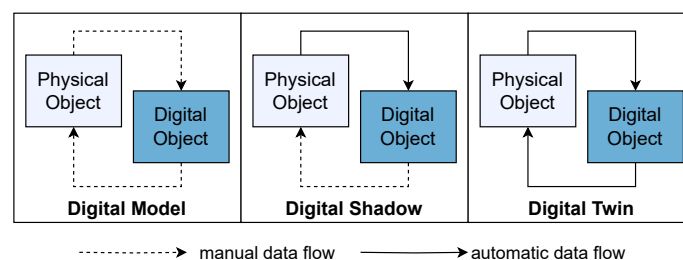


Figure 1. Differences between a DM, DS, and DT according to the classification of [23].

DTs are increasingly recognized as a foundational technology in the industrial domain, enabling real-time virtual replication of physical assets and processes to enhance operational insight and control. They support continuous monitoring, predictive maintenance, and process optimization by linking data flows bidirectionally between the physical and digital realms, thus facilitating proactive decision-making and reducing downtime [24]. Current research emphasizes their role in advancing I4.0 by integrating DTs with cyber-physical systems to create smart, adaptive production environments [25]. However, challenges persist, including managing complex data integration, ensuring model accuracy, addressing cybersecurity concerns, and developing standardized frameworks for implementation [26].

In summary, a DT is a dynamic virtual representation of a physical asset or system that enables use cases such as real-time monitoring, simulation, and optimization. It serves as a foundational technology in I4.0, facilitating improved operational efficiency.

3. Related Work

Existing research relevant to this study predominantly employs DTs as a foundational element for industrial MASs. Basically, the integration of MASs and DTs in industrial settings has attracted significant research interest, with current studies generally differentiating between two main categories: MAS with DT, where DTs serve as agent services for sensing and acting upon physical assets, and MAS for DT, where agent-based approaches are used to develop or enhance DTs [22,27].

In the first category, MAS with DT, DTs function as a service-oriented layer that enables agents to observe and influence physical processes. For example, Ricci et al. [28] propose the concept of a Web of DTs (WoDT), an open and distributed ecosystem that connects multiple DTs to support applications such as healthcare and smart mobility. Similarly, Mariani et al. [29] present an integration architecture that clearly separates the MAS layer from the DT layer, using DTs as services to access and control physical assets.

Conversely, the MAS for DT category focuses on leveraging MAS architectures to build or enrich DT frameworks, often incorporating cognitive features. Cruz-Salazar et al. [30] present an agent-based Cyber-Physical Production System (CPPS) architecture that supports self-x capabilities in industry, utilizing the Asset Administration Shell (AAS) for representation. Lorente et al. [31] discuss agent-based DT modeling for maintenance coordination, while Zheng et al. [24] integrate MAS components with semantic models within a five-dimension DT framework, where agents facilitate interaction among manufacturing elements.

Our approach aligns with the first MAS with DT category; however, it distinguishes itself by not using DTs solely as a medium to access physical resources but also to virtually represent the MAS itself. Thus, we focus on modeling run-time and design-time information of the agents within the DT to represent the structure and distributed state of the MAS. This agent-centric representation forms a foundation for enhanced transparency and inter-agent and human-agent collaboration, a perspective that remains underexplored in the existing literature.

Only a few studies have adopted a similar agent-centric DT approach. Sidorenko et al. [6] propose a capability- and skill-based framework leveraging the AAS (Asset Administration Shell) for collaboration between humans and agents in modular and reconfigurable production systems. Their DT enables agents and humans to be represented based on capabilities, with specific agents managing dynamic task assignment in mixed human-agent processes. While this study also uses skill-based modeling and AAS, the focus and technical implementation differ from our research. Sidorenko et al. focus on adaptivity in plug-and-produce production systems, whereas our work targets a DT as a shared medium to enable transparency and connectivity of agents and humans in increasingly autonomous MAS. From a conceptual point of view, the plug-and-produce production systems is realized based on a hierarchical MAS structure with managing agents for the flexible execution of processes, while we propose loosely coupled task-oriented collaborations between agents and humans based on a DT. Technically, we virtually represent agents and no humans in the MAS, specify a distinct AAS model structure, and adopt a microservice-based approach to provide control interfaces for flexible cooperation.

Furthermore, Ferreira et al. [7] introduce a supervision tool for MASs that retrieves and visualizes agent information such as hardware resources, skills, and performance metrics. This work differs from ours primarily in its limited focus on agent monitoring rather than

collaboration between agents and between humans and agents, as well as in its technical implementation. Unlike our approach, Ferreira et al. use sniffer services instead of a DT infrastructure, the JADE framework instead of a collection of microservices, and individual data models instead of the standardized AAS data model specification.

Overall, the novelty of our study lies in three key aspects: first, the use of DTs to represent and provide access not only to physical resources but to agents in the MAS itself; second, the adoption of a skill-based modeling approach for this purpose grounded in the AAS standard; and third, the conceptualization of a DT infrastructure that forms the backbone for improved transparency and seamless agent and human collaboration in increasingly autonomous industrial MASs.

4. Method

This study employs a Design Science Research (DSR) approach to develop and validate a novel technical framework that fosters transparency and collaboration between agents and humans within industrial MASs with an increasing degree of autonomy. DSR is well-suited for creating innovative artifacts that address complex real-world problems through iterative design, implementation, and evaluation [32].

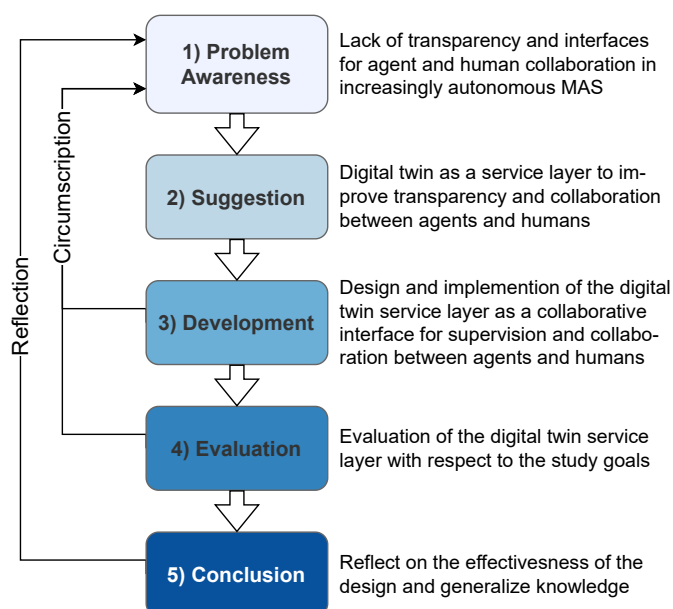


Figure 2. Research process following the multi-step framework outline by Kuechler and Vaishnavi [33].

As illustrated in Figure 2 the DSR research process of the study follows the multi-step approach from Kuechler and Vaishnavi [33]:

- Awareness of the problem:** As outlined in Section 1 and Section 2, the black-box nature of agents and lack of collaboration interfaces impair transparency and inter-agent and human-agent collaboration in increasingly autonomous industrial MASs.
- Suggestion:** In response, a novel DT framework is proposed that introduces a DT-based service layer for MASs to alleviate transparency gaps and improve inter-agent and human-agent collaboration.
- Development:** Accordingly, the DT service layer is developed as a collaborative interface to allow monitoring, supervision, and seamless inter-agent and human-agent collaboration in the MAS. First, in Section 5 a technology-independent general design

is proposed based on a comprehensive review of the existing literature, standards, and architectural principles. Thereby, a standard-based data model, a microservice architecture for managing DT data, and operational workflows are specified as core components of the novel DT framework. Second, the design is applied to the real-world industrial domain in a case study where the design is implemented as described in Section 6.

4. **Evaluation:** Next, the DT framework is evaluated in the research context of the case study. Thus in Section 7, it is examined whether the DT meets predefined design goals by deploying the service infrastructure, conducting tests, and assessing quantitative and qualitative metrics related to the functional and non-functional requirements. These have been derived from the research context of the case study and general industrial quality attributes discussed in Section 2.
5. **Conclusion:** Section 8 addresses the potential threats to both the internal and external validity of the DSR study. Finally, in Section 9 results are summarized and insights from evaluation and discussion are synthesized to generalize knowledge about the effectiveness of the design with respect to the study problem for broader industrial adoption.

In summary, this study employs the DSR methodology, adhering to the multi-phase research process described in [33]. In the course of the study, a general framework for transparency and human-agent collaboration in increasingly autonomous MAS is developed in the abstract design phase and then implemented and evaluated in a case study to assess its feasibility in a real-world industrial scenario. As a result, this study communicates the effectiveness of the proposed DT framework as a possible solution for limited transparency and inter-agent and human-agent collaboration challenges in increasingly autonomous industrial MAS.

5. Design

Structured into three subsections, this section presents the technology-independent development phase in which a general design of the DT service layer is conceived. In Section 5.1 a standard-based data model for the virtual representation of agents is specified. Subsequently, in Section 5.2, a microservice architecture is established to delineate the service components and their interactions that collectively comprise the DT service layer. Finally, operational workflows are outlined in Section 5.3 to describe how the DT service layer can be leveraged as a foundation for transparency and inter-agent and human-agent collaboration.

5.1. Data Model

This section specifies a standardized data model for the DT service layer to virtually represent agents in the MAS. For this purpose, the AAS of the Reference Architecture Model Industrie 4.0 (RAMI 4.0) is being used as a popular foundation for DTs in I4.0 [34]. The AAS provides a meta-model for representing assets in a modular and interoperable manner. By adopting the AAS as the foundational structure, this approach allows for consistent representation and interaction across heterogeneous system components while maintaining alignment with established industrial standards [35].

In the proposed model, each agent in the MAS is represented by an individual digital counterpart, which is embodied by its own AAS instance. This AAS instance encapsulates both its static design-time and its dynamic run-time information comprised into AAS submodels. Following the principles of capability-based engineering and making use of the proposed AAS ontology by Bayha et al. [36], two submodels form the core elements of the DT data model: the Capabilities submodel and the Skills submodel.

The Capabilities submodel describes the abstract goals and tasks an agent can accomplish in the context of an industrial process. Each capability represents a potential contribution to MAS-level objectives, such as production scheduling, task coordination, or quality control. Explicit modeling of capabilities within the AAS supports automated discovery of suitable agents to handle certain tasks, thereby facilitating inter-agent as well as human-agent collaboration during process execution.

The skills submodel complements this by specifying the concrete executable functions employed to realize agent capabilities. Each skill defines an operation that the agent can either execute internally or expose for invocation by other agents in the MAS. The definition of a skill includes both design-time and run-time aspects. Design-time aspects provide support for well-defined and interoperable service interfaces. They include, but are not limited to, required input variables, expected outputs, and technical meta-data to invoke agent services such as endpoint addresses or operation timeouts. The run-time aspects represent execution data, for example, the current execution state or the operation results, to support process monitoring and ensure transparency. This dual view enables the DT not only to describe what an agent can do, but also provide information on how to invoke other agents services and monitor agent-based process execution.

Together, the Capabilities and Skills submodels establish a hierarchical description linking the strategic goals of an agent to the concrete operations required to achieve them. A central characteristic of the model is the explicit linkage between capabilities and the corresponding skills that implement them. This relationship is formalized through dedicated Relationship elements, which associate one or more skills with a specific capability. In this way, abstract agent goals are systematically anchored to their concrete operational methods. On the one hand, capabilities simplify agent service invocation and mapping of agent skills to process steps. On the other hand, skills allow to track the execution of subtasks as a baseline for transparent monitoring of agent-based task execution.

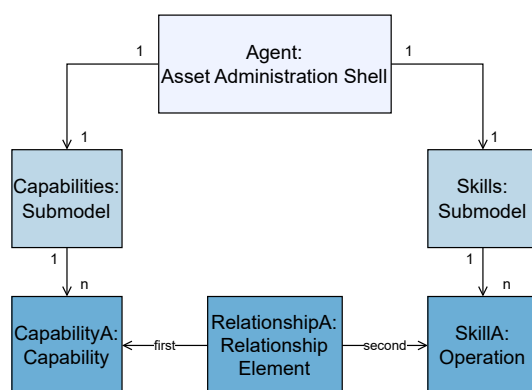


Figure 3. Structure of the AAS instance to represent agents in the MAS based on the skill-based approach discussed in [36].

The AAS structure is inherently extensible, allowing additional submodels to be incorporated depending on individual application requirements. For instance, a process submodel extension can allow dynamic mapping of agent capabilities to process entities to support flexible task-assignments in plug-and-produce system settings, as discussed in Section 3. Other examples include but are not limited to dedicated documentation submodels to provide human operators with explanatory resources or an enhancement with reasoning and decision-making aspects inspired by the Belief-Desire-Intention (BDI) model. Such extensions might promote more effective human-agent interaction and enrich the basis for collaborative problem-solving. However, the focus of this study is on the

development of a standard-based MAS representation concerned with transparency and connectivity goals. An extension of the baseline data model with additional submodels might be useful to invest in subsequent studies to support a broad range of industrial use cases but is out of the scope of this study.

Figure 3 illustrates the overall AAS representation graphically. The data model follows a layered structure in which each agent's AAS instance encapsulates both high-level objectives in the Capabilities and executable functions in the Skills submodel, together with the relational constructs to interconnect them. By providing a standardized and extensible means for describing agents within a DT, this model lays the foundation for automated discovery and orchestration of MAS participants, invocation of interoperable skill interfaces for agents and humans, and transparency of process execution for subsequent supervisory mechanisms.

5.2. Architecture

The integration of DTs with industrial MAS requires an architecture that is capable of managing agent design-time and run-time information and providing interfaces to access agent data reliably and in real-time. To meet this requirements, we propose a generic microservice-based architecture in which the DT infrastructure acts as an intermediary information system for the collaboration of agents and humans. The architecture basically comprises two subsystems, as illustrated in Figure 4: the MAS and the DT service layer. These subsystems are closely interlinked through standardized data management and interaction mechanisms to close the loop between descriptive agent data within the twin and actionable process execution in the MAS.

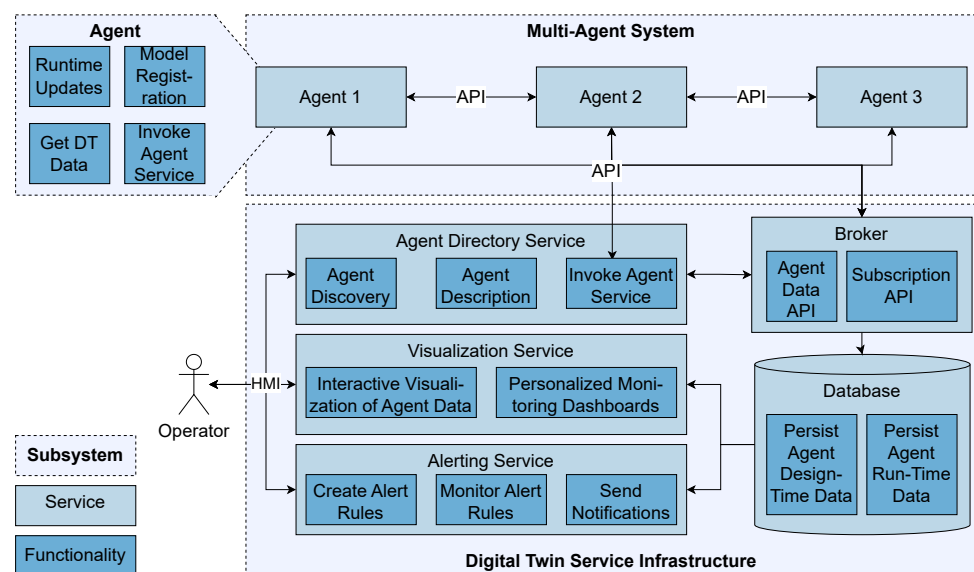


Figure 4. Architecture of industrial MASs enhanced with a DT infrastructure for increased transparency and inter-connectivity.

The first subsystem is the MAS that automates the operational tasks of the industrial plant based on agents. Agents register themselves in the DT infrastructure during initial deployment or re-deployment, so that the current AAS instance of the agent is created or updated within the twin. Whenever an agent executes or invokes an operation, real-time run-time updates are communicated back to the DT infrastructure, ensuring that execution data remain synchronized with the system state. In this way, agents do not only provide their own data to the DT but can also retrieve information about other agents from it. To retrieve information from the DT, agents can actively request data using an API or

automatically receive notifications by subscribing to events such as value updates, for example, to invoke capabilities of agents utilizing the DT. This enables agents to discover other agents in the MAS, explore which operations they expose, establish task-oriented collaborations, monitor the state of other agents, and receive notifications on the occurrence of events.

The second subsystem is the DT infrastructure, which provides the central services required to manage the agent data and APIs and HMIs as a baseline for access and control by other agents and human operators. Its architecture, following microservice principles, is composed of five core components. The first is a database, which ensures persistent storage of AAS-based agent models and run-time execution records. The second is a broker service, which operates as the access layer and manages interfaces through which both agents and humans can actively read and write DT data or create subscriptions to receive agent data updates automatically. The broker also regulates real-time data exchange and enforces structural consistency of AAS representations. The third component is a visualization service, responsible for rendering both agent design and run-time information graphically. This component supports supervisory operators in understanding current MAS interactions, monitoring performance, and diagnosing potential issues. The fourth component is an alerting service, which allows to flexibly create alerting rules and issues notifications on channels such as e-mail or messaging apps to warn in case of abnormal runtime conditions such as execution failures, timeouts, or contradictory agent states. The final component is an agent directory service that allows operators to graphically discover agents in the MAS, explore their AAS representations, and possibly invoke agent operations for automated task execution.

Although the proposed DT architecture focuses on improving transparency and connectivity in industrial MAS only, it is extensible by design. For instance, supplementary security and interoperability mechanisms can be incorporated through the integration of additional services such as identity providers or API gateways to enable restricted access and secure interaction across organizational boundaries based on authentication and authorization services. Recent research from Miadowicz et al. [37] highlights the value of integrating such mechanisms and proposes how to flexibly integrate different types of functional modules into a comparable architecture of an I4.0 middleware platform.

In summary, the architecture of the DT service layer provides an information-centric, microservice-based bridge between industrial agents and human supervisors. On the one hand, agents provide design-time and run-time information by registering themselves at the system and continuously update dynamic runtime information. On the other hand, agents and human operators benefit from the DT service layer. Agents can use an API for agent discovery, inter-agent collaboration, and monitoring of the MAS. Operators are supported based on three services that provide graphical interfaces to supervise and control agent-based process execution.

5.3. Workflows

The DT service layer introduces several features for inter-agent and human-agent collaboration including agent discovery, description, monitoring, alerting, and service invocation. These features basically enable three operational workflows that define the key interaction mechanisms with the DT service layer: agent registration, agent collaboration, and agent supervision.

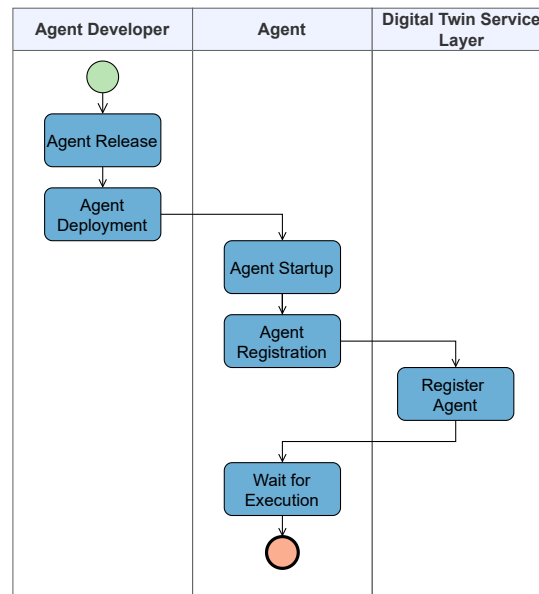


Figure 5. Agent registration process on initial deployment or update of the agent to virtually represent agents in the MAS.

The agent registration process, as depicted in Figure 5, occurs during the initial deployment of an agent or its update to a new version. In this step, the agent publishes or updates its AAS-based model in the DT infrastructure, including meta-information, declared capabilities, and available skills. In this way, the registration phase establishes transparent, up-to-date agent descriptions, enabling both human operators and other agents to find the agent, assess the functional role and current state within the MAS, and possibly invoke its services. Not illustrated, but similarly to the registration process, there is an agent unregistration process where agent models or certain services are unregistered from the DT as they are currently not available or should not be used anymore for future plant operation.

The second workflow defines interaction patterns for agent collaboration on the basis of agent discovery and service invocation mechanisms, as illustrated in Figure 6. After registration, agents and humans can query the DT to identify agents available across the system using an API or HMI. Discovery queries may target all registered agents or filter according to specific capabilities, for instance, identifying which agents can supply a given material, schedule a production task, or conduct quality testing. Since the agent data model also specifies technical details on how to use and access agents endpoints, agents and humans can invoke agent services directly through an API or HMI or use subscription and alerting mechanisms to automatically execute agent capabilities on the occurrence of events. By embedding such endpoint information within the twin, the architecture establishes a seamless bridge between descriptive AAS data and actionable process execution. This streamlines distributed process orchestration, as an agent may autonomously discover and invoke operations of other agents.

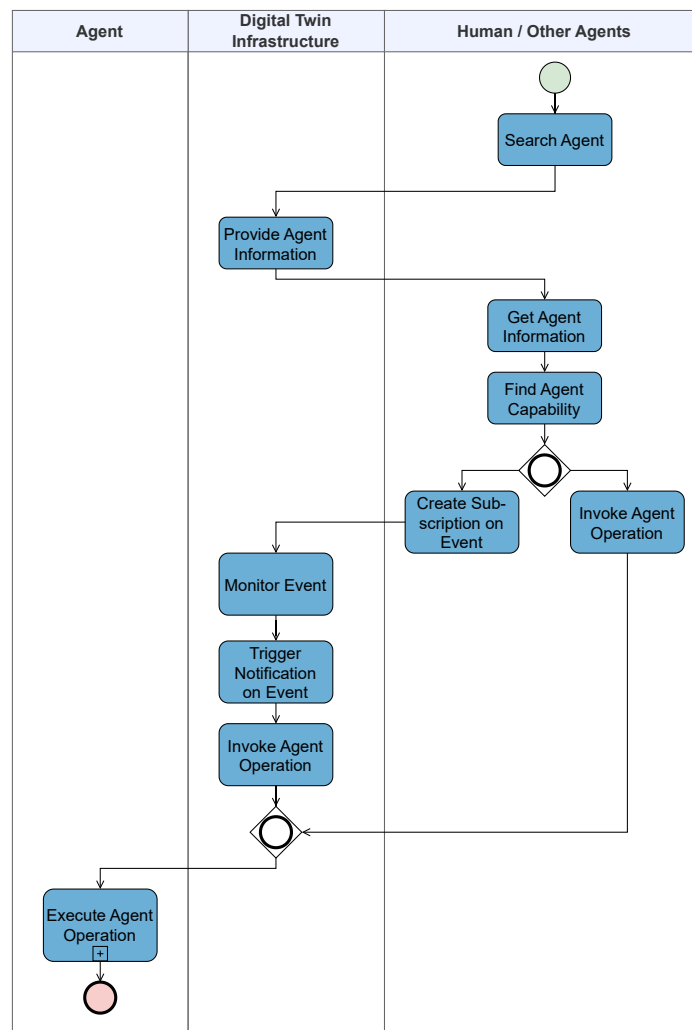


Figure 6. Agent collaboration workflow based on discovery and active or event-based interaction mechanisms for the automation of processes.

The modular approach and strict separation of components into the MAS and DT layer in the proposed framework, enables a flexible synchronization of agents and their digital representation as a DS or DT according to the classification of [23] discussed in Section 2.3. Following the DS approach, agents or humans can invoke agent services directly using the agents' API or an HMI. In this scenario, the information flow is one-directional from the physical object to the digital one, as the capabilities are directly called from agents or humans without changing the data model in the DT. Following the full DT approach, agents or humans can update the data in the DT instead of calling agents directly. In this case, subscription mechanisms can be used to invoke agent capabilities automatically on value update for a bi-directional synchronized representation of the physical and virtual object. The implementation of a DS or DT has both advantages and disadvantages. The DS supports loose coupling, reduced dependency on the DT broker and less communication overhead. The full DT approach fosters a more reliable synchronization of software agents with their virtual representation as the digital counterpart is changed before the agent service is invoked by event-based mechanisms. Thereby, the DT ensures a consistent digital representation and synchronization of task-oriented collaborations between agents and humans.

The third workflow describes supervision mechanisms including agent monitoring and alerting features, as visualized in Figure 7. A task can either be executed by an agent or automatically invoked by event-based mechanisms of the DT service infrastructure as discussed in the previous text. During task execution, each agent continuously updates its runtime information within the DT infrastructure, including intermediate task states and final operation results. This ensures that the overall process state is transparently captured and accessible at all times. Other agents and human operators can observe these execution data either through an API or via a supervisory dashboard that visualizes the real-time operation status. Beyond monitoring, the architecture also provides subscription and alerting features to complement active monitoring strategies with event-based mechanisms for improved resilience and responsiveness within the MAS. Notifications can be automatically triggered whenever events or critical issues occur, such as an exceeded threshold or failed termination of an agent operation. Subsequently, the notification can, for example, trigger automatic adaptation of the agent or send notifications to operators to request human intervention.

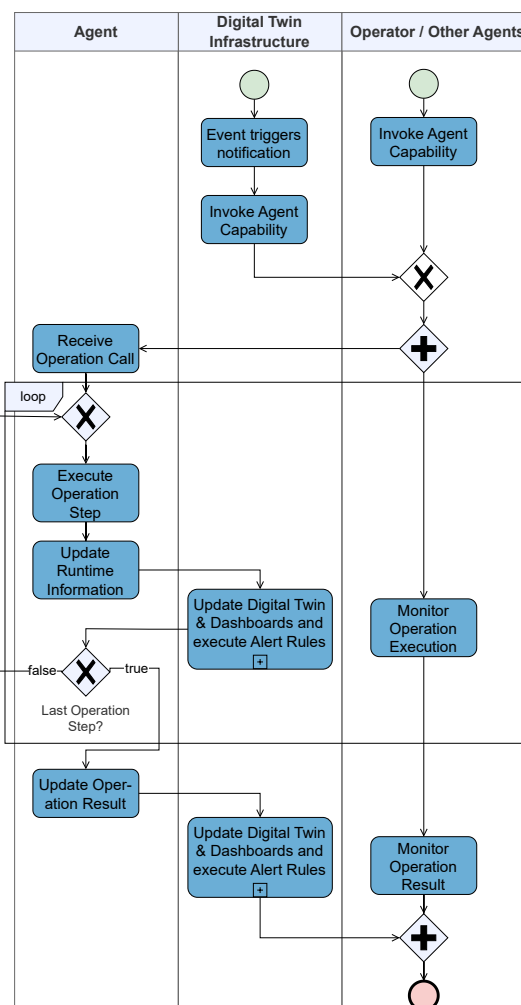


Figure 7. Agent supervision workflow that illustrates continuous synchronization of agent run-time information with the DT as a baseline for monitoring process execution and pro-active alerts in case of operational events.

Taken together, these three workflows define a continuous loop of agent integration, collaboration, and supervision within the DT environment. The registration workflow provides accurate up-to-date digital representations, the collaboration workflow enables dynamic agent discovery and flexible service invocation, and the supervision workflow ensures that agent-based process execution can be monitored transparently while issues are automatically reported to subscribers.

6. Case Study

This section presents the second development phase of the study which applies the abstract design to a real-world industrial scenario in a case study. The first subsection describes the research context of the industrial scenario in a Concentrated Solar Thermal (CST) plant and the technical system infrastructure before the DT framework is implemented, thereby emphasizing already existing operation agents as building blocks of the MAS. The second subsection explains the implementation steps of the proposed framework by specifying the data model for agents in the MAS and realizing the proposed architecture on the basis of open-source and self-developed microservices.

6.1. Industrial Scenario

The case study is conducted in the context of a semi-autonomous CST plant in the German city of Jülich, designed to generate thermal energy either for electricity production or for direct industrial process use [38]. The plant comprises about 2.000 mirrors that concentrate sunlight onto a central receiver on the solar tower. At the receiver the concentrated irradiation is absorbed and thermal heat is subsequently converted into electricity or directly used in processes of interconnected industrial facilities. To ensure a reliable energy supply even in periods of low or no solar irradiation, a heat storage is integrated for effective thermal energy storage and continuous energy supply [39].

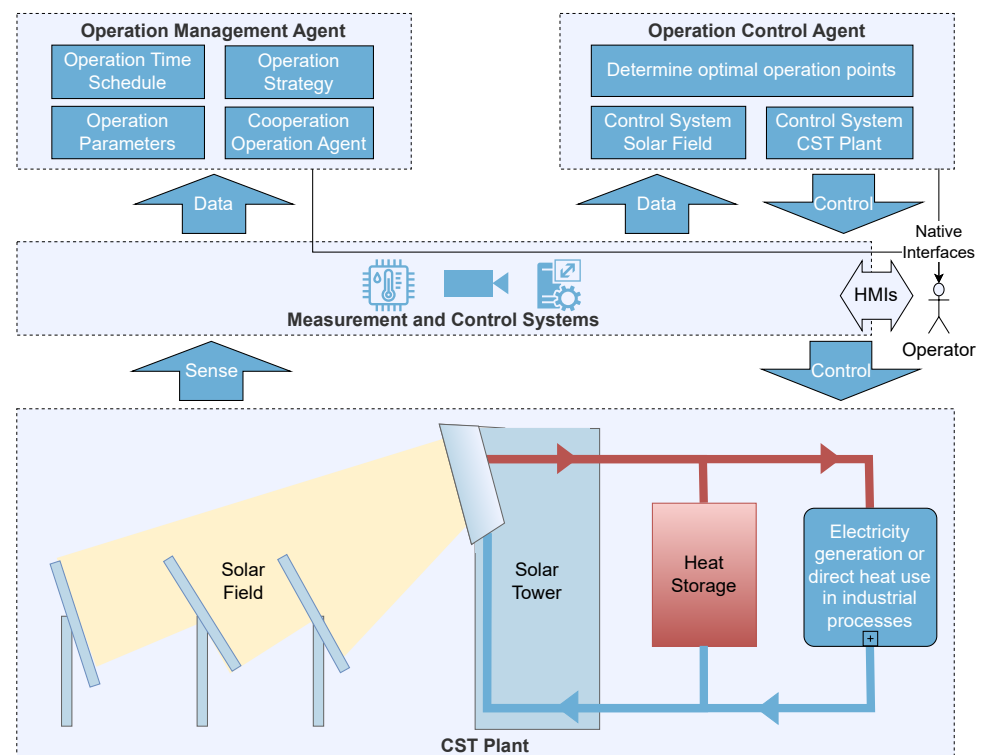


Figure 8. Illustration of the case study scenario in the context of a semi-autonomous CST plant.

Within this setup, two agents automate tasks of the CST plant operation process as illustrated in Figure 8:

1. **Operation Control Agent:** The operation control agent analyzes data from an infrared and a flux density camera, a solar irradiation sensor, and control systems for the heliostat and CST plant. Based on the input data, the operation agent determines optimal operation points and uses interfaces to control systems to regulate the mirrors in the solar field and plant components in the solar tower. The goal of the operation control agent is to produce heat according to given target parameters such as the target temperature and mass flow despite dynamic environmental conditions.
2. **Operation Management Agent:** The operation management agent leverages weather forecasts, actual weather measurements, process requirements for electricity production or industrial heat generation, and the operational states of the plant components to parameterize a simplified CST plant model. Its goal is to compute an optimized operating schedule, including start/stop times, strategic decisions for power generation or storage, and target parameters for operation within actual environmental conditions. The finalized operation plan is then shared with a human operator as a recommendation for optimized operation.

Currently, both agents automate single tasks but do not collaborate to carry out the CST plant operation process autonomously. Instead, one agent creates an optimized operation plan as a recommendation for human operators, and the other agent is able to automate plant operation according to given target parameters that have to be provided by a human operator. In between is the human operator who receives an operation plan recommendation from operation management agent and decides how to execute the operation plan by configuring the operation control agent accordingly using their native custom-built and distributed agent interfaces. With respect to agent and human collaboration, the baseline scenario lacks a consistent self-description of agents, unified discovery and collaboration mechanisms, and simplified interfaces for inter-agent and human-agent interaction. This limits the flexible interconnection of intelligent agents and humans to form dynamic task-oriented collaborations for the end-to-end automation of the CST plant heat generation process.

Furthermore, the agents in the MAS function as black boxes with no explicit interface to other agents and human operators besides distributed system logs of the agent software and single user interaction points, for example, to share operation plans in case of the operation management agent. Additionally, human operators and agents can monitor the result of agent-based process execution from the central control room indirectly through the use of HMIs of measurement and control systems to assess if sensor measurements are within the permitted range and set points for industrial components are configured appropriate in relation to the environmental conditions. At the moment, there are no possibilities to receive live-insights of the MAS including the availability of agents and their execution states. Therefore, it is not transparent which tasks are currently performed by agents and how they perform the task with respect to functional correctness and operational efficiency. Due to the lack of transparent agent run-time information and availability of appropriate monitoring and control interfaces to observe and possibly intervene agent-based task execution, supervising MAS remains a challenging task for both agents and human operators.

In summary, the case study is conducted in a semi-autonomous CST plant. The process for heat generation is partly automated based on an operation management agent and an operation control agent which create an optimized operation schedule and execute plant operation according to given target parameters. In between is a human operator who makes the final decision on the proposed operation plan and controls and monitors agent-

based plant operation accordingly. Currently, both agents operate as black-boxes without interfaces to humans to supervise agent-based task execution and effectively collaborate with agents in mixed human-in-the-loop processes.

6.2. Implementation

Next, the implementation of the DT service infrastructure in the industrial scenario is discussed encompassing the specification of the AAS data model and realization of the service infrastructure.

In alignment with the data model specification, each agent within the MAS is assigned its own AAS representation. Thus, the data model for the case study encompasses two AAS instances, corresponding to the operation control agent and the operation management agent as illustrated in Appendix A1 and Appendix A2. The Capabilities submodel enumerates the tasks attributable to each agent within CST operation. The Skills submodel identifies the functionalities required to fulfill these capabilities. For instance, the operation agent possesses skills for autonomously initiating, controlling, and safely shutting down plant operations, including necessary sub-skills for system startup, heating, and cooling. The planning agent, in turn, is endowed with capabilities for daily operations planning, encompassing schedule and strategy recommendation, parameter computation, and operation plan recommendations.

To implement the artifact, additional components are integrated into the MAS layer, and the DT service infrastructure is incorporated as a supportive layer for the collaboration of agents and human operators in the MAS as illustrated in Figure 9. In the MAS layer of the system, both agents are enhanced with a generic agent-proxy module¹. This module facilitates the integration of agents with the DT service layer by providing features for the registration of the agent model, the provision of real-time updates, and seamless data exchange with the DT service infrastructure. Furthermore, the module exposes an REST API, enabling the unified invocation of the agent operation by other MAS entities or human operators over the network. By extending single agents with an isolated agent-proxy module, the design ensures modularity and extensibility for the easy integration of both existing and new possibly heterogeneous software agents without modification of the original agent logic.

¹ Open-source software implementation published by the authors at: <https://github.com/DLR-SF/Agent-Proxy>

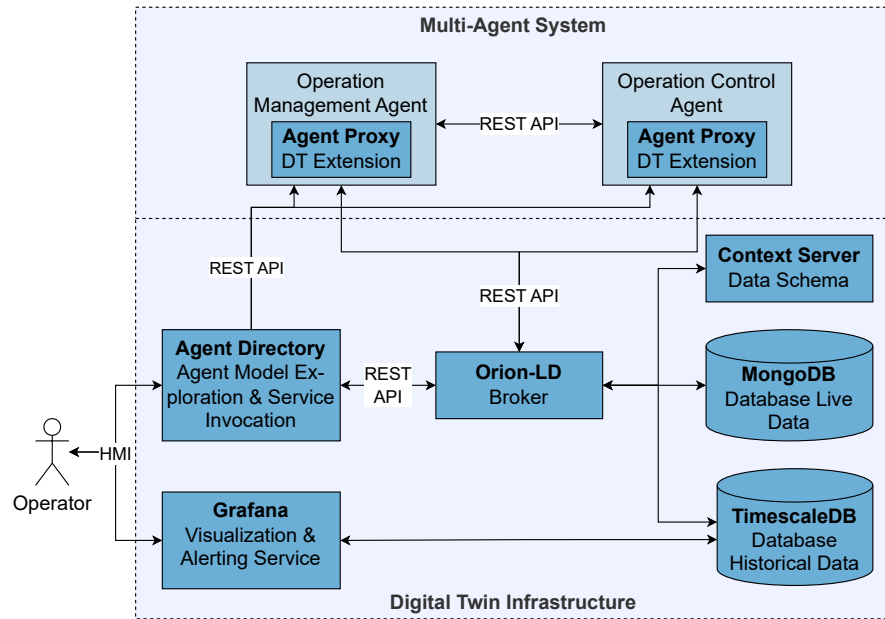


Figure 9. Implementation of the generic architecture in the context of a semi-autonomous CST Plant.

The implementation of the DT infrastructure builds upon a composition of open-source software. The broker is implemented using FIWARE technology, leveraging Orion-LD for agent data management². Requests to Orion-LD are validated against the AAS data schema stored in the context server³. Data is managed using a document-based MongoDB for live data⁴ as well as TimescaleDB for historical data⁵. Visualization and operational monitoring are provided through Grafana, which enables the creation of interactive dashboards, system health checks, and flexible alerts and notifications for plant operators⁶. Additionally, an agent directory service has been developed and published as open-source software by the authors, so that operators can explore agents in the MAS and invoke their services⁷.

Opting for Orion-LD as a DT broker natively supports desired functionalities; however, it prescribes the use of Next Generation Service Interface-Linked Data (NGSI-LD) as both the information model and API for publishing, querying, and subscribing to DT data. NGSI-LD and AAS are distinct standards for DTs, with differences in semantics and syntax, but AAS structures are compatible with NGSI-LD, and community models already exist to represent AAS structures with NGSI-LD⁸. The data model — whether in NGSI-LD or AAS — can be interchanged using translation services or serialized intermediary formats like JSON, OPC UA, or RDF, guaranteeing technical interoperability between both standards [40,41].

Based on the infrastructure introduced, the level of automation of the operation process in the CST plant can be easily increased in the heat generation process based on the continuous task-collaboration of both agents. Thereby, the operation management agent invokes the operation control agent directly to start or stop operation of the plant according to its operation schedule with optimized start and stop times. Furthermore, the optimized operation parameters of the operation management agent are stored in the DT as a result of

² Open-source software implementation available at: <https://github.com/FIWARE/context.Orion-LD>

³ Open-source software implementation available at: <https://nginx.org>

⁴ Open-source software implementation available at: <https://www.mongodb.com/>

⁵ Open-source software implementation available at: <https://github.com/timescale/timescaledb>

⁶ Open-source software implementation available at: <https://grafana.com>

⁷ Open-source software implementation published by the authors at: <https://github.com/DLR-SF/Agent-Directory>

⁸ Community data model to represent the AAS meta-model with NGSI-LD: <https://github.com/smart-data-models/dataModel.AAS>

the plan operation capability. Every time operation parameters are changed according to a changed strategy or environmental dynamics, a notification is triggered by a subscription that invokes the capability of the operation control agent to adopt its target parameters during operation. By storing the output of the operation management agent in the DT and automatically invoking the operation control agent, the whole heat generation process of the CST plant is autonomously conducted by the MAS based on the collaboration of two agents. The human operator takes over a supervisory role in this scenario, where the operator monitors agent-based process execution, receives alerts in case an intervention is necessary, and generally supervises the operation process. Compared to the initial baseline situation where a human received an operation plan as a recommendation from operation management agent and needed to configure and re-configure the operation control agent for conducting the operation appropriately within the defined thresholds according to environmental conditions, the DT frameworks enables a seamless integration of agent and human actors for the flexible end-to-end automation of processes.

In summary, the DT framework for the inter-agent and human-agent collaboration is implemented by creating an AAS model for each agent in the system, realizing the DT service layer based on a composition of open-source tools, and integrating agents with the DT service layer based on enhancement of agents with an agent-proxy module. The heat generation process can now be fully automated using agents, with a human operator remaining as the supervisor of the system with monitoring and higher-level control functions.

7. Evaluation

The evaluation aims to determine how well the DT infrastructure provides transparency and seamless collaboration between humans and agents in an increasingly autonomous MAS. To assess whether the DT meets the predefined design goals, the fulfillment of functional and non-functional requirements is examined in the context of the case study.

As a basis for evaluation, a list of functional and non-functional requirements has been derived based on the study goals discussed in Section 1, an examination of the case study scenario discussed in Section 6.1, and the consideration of general relevant quality attributes and acceptance criteria for industrial systems from the recent literature as discussed in Section 2. For each requirement, quantitative or qualitative metrics have been defined to measure the effectiveness and efficiency of the design. Tests are then conducted in a 30 hours test period in CST plant heat generation processes that are monitored and controlled based on the collaboration of agents and humans and actively supported by the DT infrastructure as described in Section 6.2. During the experiments metrics are measured and results are compared to the defined acceptance criteria. In this context, the following text describes the results of the evaluation from a functional and non-functional requirements perspective.

The evaluation from a functional requirements perspective investigates whether the DT infrastructure provides all the desired capabilities for improved transparency and agent and human interaction within the MAS as summarized in Appendix A1. With respect to transparency goals, the infrastructure is able to represent agent design and runtime information including capabilities, skills (A1), and runtime states and (A2) provides an API for agents (A3) and an HMI for humans (A4) to access agent data. The infrastructure also supports inter-agent collaboration, examining the ability to discover agents in the MAS (A5), understand and invoke their services (A6), monitor their states actively (A7), and receive timely notifications about relevant events (A8). Regarding inter-agent interactions, these features enable flexible task-oriented collaborations and different self-x properties of the system including self-description, self-discovery, self-monitoring, self-assessment, and

self-organization of task-oriented collaborations. Moreover, human operators get additional services to discover agents in the MAS (A9), execute or interrupt agent operations (A10), visualize agent run-time data for monitoring of agent-driven processes (A11), and receive important alerts when intervention is required (A12). Based on these features, human operators gain transparency over MAS structures and state as well as interfaces to supervise agent-based process execution. Hence, from a functional perspective, the capability-based AAS representation of agents in the MAS and the microservice-based DT service layer provide a powerful foundation for advancing collaborative and transparent agent-based process execution in the scenario.

Next, the evaluation from a non-functional requirements perspective focuses on main quality attributes to assess the suitability and utility of the DT artifact for the industrial use case. Thereby, the standardized data model and the implementation of the DT infrastructure is assessed based on quantitative and qualitative metrics. For each of the quality attributes either quantitative metrics are measured or discussed from a qualitative perspective in comparison to the initial baseline scenario to evaluate if the artifact meets the demands of modern production systems. Appendix A2 summarizes the results in tabular form.

First, the evaluation of the standardized data model is discussed with respect to its interoperability, completeness, and extensibility:

A12. Interoperability (A13): There has been no explicit data model in the baseline scenario described in Section 6.1 but distributed, unstructured, and partly indirect data on MAS structure and operation state in external documentation of the agents and native agent interfaces. The data model of the contribution is enriched with meta-data for an accurate and interoperable representation of the MAS, aligned with the widely adopted AAS standard for industrial DTs. The AAS standard accommodates a variety of industrial communication protocols and supports multiple serialization formats, including JSON, XML, and RDF. Its use of standardized semantic vocabularies and vendor-neutral structures ensures that asset information can be reliably exchanged, integrated, and understood across diverse platforms and organizations. Although the implementation of the DT infrastructure in the case study based on open-source software services from FIWARE may require the use of translators for seamless technical interoperability of NGSI-LD and AAS, the underlying structural and semantic compatibility remains strong, as detailed in Section 6.2. Next to the interoperability across organizational borders, the AAS standard also provides flexibility to generally choose other services for the implementation of the DT service or exchange services in the DT infrastructure layer with other technical solutions than proposed in the case study. The broad acceptance of the AAS standard in the industrial domain results in the availability of different software solutions that support to manage AAS-based DTs and provide access to their data [35].

A13. Completeness (A14): In contrast to the baseline scenario, where design-time information is only available via external documentation and run-time insights were gleaned indirectly from various monitoring and control systems or custom-build, distributed, and native agent interfaces, the AAS-based model consolidates both design-time and run-time data into a unified, structured format. Facilitating the skill-based ontology as suggested in [36] the format provides transparent insights to agent capabilities and runtime data for monitoring and automation of industrial processes. In terms of completeness, the model is capable of representing all required aspects of agent self-description, employing a combination of AAS, Submodel, Capability, Relationship, and Operation entities. Each entity comprises between 10 and 30 properties that cover both design-time and operational attributes.

- A14. **Extensibility (A15):** Compared to unstructured data in the initial baseline scenario which has no build-in extension mechanisms, the contributed data model demonstrates significant extensibility. It readily supports the addition of new data entities or properties through the introduction of new submodels or by leveraging the extension mechanisms embedded within the AAS metamodel as described in Section 5.1. This allows enhancing implemented data models with additional information in a flexible way accommodating evolving system requirements and emerging use cases.

Second, the evaluation of the implemented DT infrastructure is examined from a non-functional perspective with respect to performance, robustness, and adaptability:

- A15. **Performance (A16):** On average, read and write requests to DT Service infrastructure take about 40ms, which is significantly below the accepted 100 ms threshold for real-time performance in the industrial domain [42].

- A16. **Robustness (A17):** In the baseline scenario, each agent has been deployed on a server as an isolated software agent without redundancy or load balancing mechanisms. Even though there exist no numbers on the robustness of the MAS without DT service layer, robustness might be affected by hardware and software faults as well as limited monitoring capabilities. During the evolution with the DT service layer, there were no errors during the 30-hour evaluation period, indicating the functional correctness of the first version of the infrastructure. Although a longer test period may uncover functional issues, the usage of a microservice-based architecture allows to deploy the implementation on a cluster out of five servers and configure replication for each microservice based on load individually. This enables reliability of the DT infrastructure even in the event of failure of the underlying hardware or dynamic workloads. Moreover, monitoring capabilities and proactive alerts provide functional features to ensure the robustness of heat generation processes.

- A17. **Adaptability (A18):** Finally, adaptability is validated by the capability of the solution to seamlessly add or remove agents from the MAS or flexibly integrate them for the gradual transition towards higher degrees of autonomy. While the initial baseline situation provides no specialized adaption mechanisms, the contribution introduces a loosely coupled network of agents, where new agents can easily register on the system so that other agents and human operators can find the agents and create task-oriented collaborations utilizing DS or DT interaction patterns. Thereby, agents and humans can simply search for needed capabilities and dynamically discover and invoke agents that are able to provide these services without the need for software modification. In the same way, an agent can unregister itself at the DT infrastructure so that other agents and human operators know that the agent services are no longer available. In this case, they might need to use the services of other agents if available or inform human operators to do manual tasks using notification services. In the context of the case study, adaptability has especially been shown by flexibly integrating both agents for full automation of the heat generation process while the human operator takes over a supervisory instead of an active role in the overall process. Furthermore, the DT twin framework lays the foundation for a flexible integration of additional agents for the automation of further industrial process tasks.

In summary, the DT infrastructure provides transparency and supports collaboration between agents and humans in increasingly autonomous MAS. From a functional perspective, the service layer enables self-x capabilities of the MAS including self-description, self-discovery, self-monitoring, self-assessment, and self-organization of task-oriented collaborations. Furthermore, the infrastructure enables human supervisory features en-

compassing an agent directory, monitoring, and alerting service. A first evaluation of the DT infrastructure quality illustrates the benefits of the proposed solution with respect to its interoperable and extensible data model as well as its real-time, robust, and adaptable service implementation as a foundation for agent-based automation of industrial processes.

8. Discussion

This Section discusses the threats to internal and external validity of the study to transparently communicate possible limitations of the generalization of the results.

Starting with threats to internal validity, the study was conducted within the boundaries of a single case study in a semi-autonomous CST plant with a specific set of agents and operational contexts. Although implementation measures and evaluations were rigorously performed over a 30 hour test period, the evaluation primarily assessed the availability of a set of functional requirements and several core quality attributes over a limited test period. Despite the systematic approach, there remains a possibility that certain agent behaviors, rare error conditions, or unusual edge cases were not encountered or fully exercised and that metrics may have been impacted by subjective interpretation. The primary focus of this research is to perform an evaluation of the MAS with DT approach in relation to core study objectives based on the development of a novel DT framework and evaluation of its effects. Enhancing the study with a long-term evaluation and additional quantitative benchmarks would allow broader data collection and enable a more accurate assessment of additional quality attributes beyond those addressed in the initial test.

A threat to external validity is the generalization of the study results, as the evaluation took place in a specific CST plant context in a single case study environment. Applying the framework to other industrial scenarios could reveal further functional or non-functional requirements that might enrich and extend the DT framework. Although designed with adaptability in mind, further tests beyond the process-based smart CST plant, might also reveal different technological requirements for the integration of the proposed DT service layer into real-life industrial environments. To overcome these limitations, future research should consider application of the framework across diverse industrial domains, agent configurations, and operational conditions to broaden the scope and strengthen the external validity of the results.

In summary, while the study offers a thorough, multi-dimensional first evaluation of the DT service layer in a real-world setting, threats to internal validity stem from contextual constraints, limited testing period, and subjective measurement bias. The primary intent here was to establish a technical foundation and assess the feasibility and requirement satisfaction of the artifact in a real-world industrial environment. Long-term and multi-context assessments, focused on additional quality attributes such as scalability, usability, and security, are recommended for future investigations.

9. Conclusion

As a result of the DSR process, the effectiveness of the design with respect to the study goals is reflected and general knowledge on DTs as a service infrastructure is communicated. Thereby, it is assessed whether the DT service infrastructure successfully addresses challenges related to transparency and inter-agent and human-agent collaboration in increasingly autonomous industrial MAS. Furthermore, lessons learned are reflected to generalize the learnings of the approach for future investigations on transparency and collaboration in agent-based systems.

In the course of the study, a DT framework has been proposed to address the lack of transparency and collaboration interfaces in increasingly autonomous MAS facilitating the MAS with DT approach as discussed in Section 3. The framework suggests to introduce a

DT service layer as an intermediate artifact to virtually represent the design and run-time information of the agents and provide unified access to monitor and control agent-based process execution for both agents and humans. The framework comprises a standardized skill-based data model, a modular microservice-based architecture, and workflows to define key interaction mechanisms for the inter-agent and human-agent collaboration.

While the generic design of the DT service infrastructure suggests a technology-independent abstract solution concept, the case study in a CST plant allows to implement and evaluate the solution concept in a complex real-world industrial scenario. Based on the case study, the technical feasibility of the framework has been demonstrated and the quality of the solution has been assessed with respect to the main quality attributes relevant for (semi-)autonomous industrial MAS, including performance, adaptability, robustness, extensibility, completeness, and interoperability. Although an in-depth evaluation of the proposed solution in a multi-case study is necessary to evaluate the overall efficiency of the approach for a broad adoption in other industrial application fields, the complex scenario of the case study in a CST plant has proven that the DT framework successfully addresses the lack of transparency and collaboration interfaces paving the way for increasingly autonomous MAS.

Overall, the proposed DT framework successfully provides a design-based solution to overcome the black-box nature of agents and interconnect humans and agents in increasingly autonomous industrial MAS. An introduction of a DT service layer can enable self-x capabilities of the MAS for the seamless collaboration of agents and humans as a baseline for the gradual transition towards adaptive and autonomous industrial systems. Thereby, the framework incorporates self-description, self-discovery, self-monitoring, self-assessment, and self-organization capabilities to ensure efficient, robust, and flexible system coordination, supervision, and interaction mechanisms between agents and humans for the stepwise automation of processes.

Author Contributions: Conceptualization, I.M., M.K. and D.MQ.; methodology, I.M. and M.F.; software, I.M. and M.K.; validation, I.M., M.K. and D.MQ.; formal analysis, I.M.; investigation, I.M., M.K. and D.MQ.; resources, D.MQ., R.P., M.F.; data curation, I.M. and M.K.; writing—original draft preparation, I.M.; writing—review and editing, I.M. and D.MQ.; visualization, I.M.; supervision, D.M.Q., R.P. and M.F.; project administration, I.M. and D.MQ.; funding acquisition, D.MQ. and R.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Software solutions used in the study are either open-source or have been published as open-source by the authors. Links to an external repository or webpage of the institution to access the code can be found in the footnotes of the article.

Acknowledgments: We thank all DLR employees who directly or indirectly contributed to the autonomous operation of the CST plant. Special thanks go to all employees at the CST plant in Jülich for their support in plant operation, maintenance of measurement and control systems, and help to understand processes and technical system infrastructures in the CST plant as a baseline for the case study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAS	Asset Administration Shell
BDI	Belief-Desire-Intention
CPPS	Cyber-physical Production System
CST	Concentrated Solar Thermal
DM	Digital Model
DS	Digital Shadow
DSR	Design Science Research
DT	Digital Twin
HMI	Human-Machine Interface
I4.0	Industry 4.0
MAS	Multi-agent System
NGSI-LD	Next Generation Service Interface-Linked Data
RAMI 4.0	Reference Architecture Model Industrie 4.0

Appendix A

827

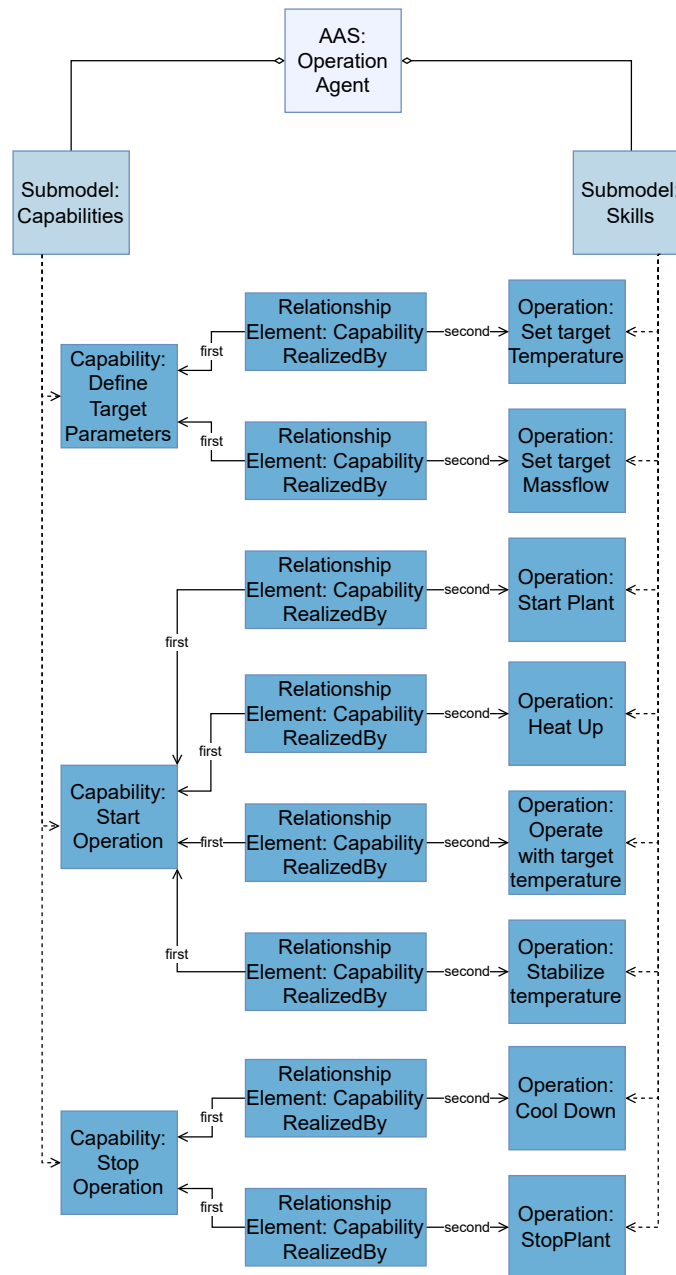


Figure A1. AAS-model instance of the operation control agent in the context of the CST plant

Appendix B

828

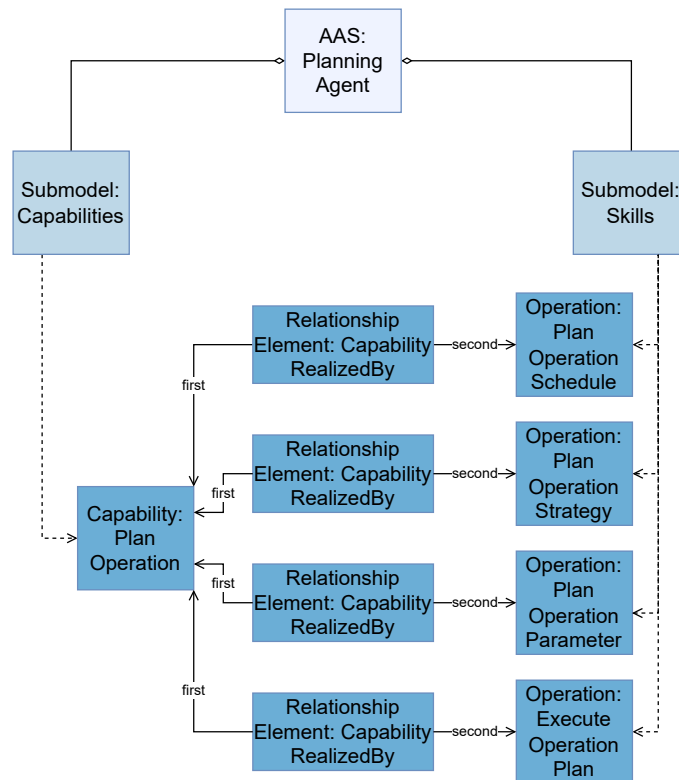


Figure A2. AAS-model instance of the operation management agent in the context of the CST plant

Appendix C

829

Table A1. Assessment of the functional completeness of the DT infrastructure in the CST plant.

ID	Requirement	Study Goal	Evaluation Method	Feature	Result
A1	Agent representation of static design-time information	Transparency	Feature availability	Data model captures agents capabilities, skills and further metadata	Available
A2	Agent representation of dynamic run-time information	Transparency	Feature availability	Data model has fields to capture agent run-time states and operation results	Available
A3	Interface to access agent data usable by agents	Transparency	Feature availability	REST API to access agent data	Available
A4	Interface to access agent data usable by humans	Transparency	Feature availability	Agent directory service as HMI to access agent data	Available
A5	Discovery mechanisms to find agents in the MAS usable by agents	Inter-Agent Collaboration	Feature availability	REST API to search for agents in the MAS and explore their capabilities	Available
A6	Interface to invoke agent services usable by agents	Inter-Agent Collaboration	Feature availability	REST API to invoke services over the network according to AAS model specification using the agent-proxy extension	Available
A7	Monitoring of agent-based task execution usable by agents	Inter-Agent Collaboration	Feature availability	REST API to actively query agent run-time information	Available
A8	Automatically receive notifications on events	Inter-Agent Collaboration	Feature availability	Subscription API to subscribe to events and receive notifications automatically	Available
A9	Discovery mechanisms to find agents in the MAS for humans	Human-Agent Collaboration	Feature availability	Agent directory service to visualize all agents in the MAS	Available
A10	Mechanisms to Invoke services of agents or stop agent operation execution	Human-Agent Collaboration	Feature availability	Agent directory service to invoke or stop running agent operations	Available
A11	Visually monitor agent run-time information	Human-Agent Collaboration	Feature availability	Visualization and dashboards to monitor agent-based process execution for humans	Available
A12	Possibility to automatically receive notifications on events for humans	Human-Agent Collaboration	Feature availability	Alerting mechanism to configure alerts and get notified automatically	Available

Appendix D

830

Table A2. Evaluation of Non-Functional Requirements of the DT Infrastructure.

ID	Requirement	Artifact	Evaluation Method	Comparison and/or Measurements	Result
A13	Interoperability	Data Model	Qualitative	<u>Baseline:</u> Unstructured data in external documentation and possibly provided by native agent; <u>Contribution:</u> Structured AAS representation as a vendor-neutral standard with metadata, various communication protocols, different serialization formats, semantic vocabularies, broad acceptance, and available software services from the community;	High interoperable standard-based data model
A14	Completeness	Data Model	Qualitative and Quantitative	<u>Baseline:</u> Distributed information in unstructured sources possibly incomplete; <u>Contribution:</u> Data model captures design-time and run-time data as well as additional meta-data; X entities and fields in total for two agents in the use case study;	Comprehensive agent model
A15	Extensibility	Data Model	Qualitative	<u>Baseline:</u> No in-build extension mechanisms so that a modification of software is necessary; <u>Contribution:</u> The standardized data model is extensible by design, providing build-in support for adding additional submodels or enhancing existing ones through AAS extension mechanisms;	Extensibility by design
A16	Performance	DT infrastructure	Quantitative	<u>Baseline:</u> Monitoring and control systems in the industrial domain generally receive a response time below 100 ms as real-time performance; <u>Contribution:</u> The DT service layer has a response time of about 40 ms for read and write requests on average;	Real-Time Performance
A17	Robustness	DT infrastructure	Qualitative and Quantitative;	<u>Baseline:</u> A server for each agent without redundancy or load balancing mechanisms; <u>Contribution:</u> Scalable microservice architecture with adaptable load balancing and redundant service deployment to improve robustness; Failure Rate 0 Errors / 30 hours; Level of Redundancy 5 Servers and 3 Service Replicas;	Horizontal scalable robustness
A18	Adaptability	DT infrastructure	Qualitative	<u>Baseline:</u> Isolated software agents which have to be actively introduced, integrated and invoked by operators; <u>Contribution:</u> Self-registration, self-description, self-discovery, and self-organization mechanisms to flexibly add or remove agents from loosely coupled network of agents in the MAS or form task-oriented collaborations with agents and other humans;	Highly adaptable MAS cooperation and collaboration structures

References

1. Estrada-Jimenez, L.A.; Pulikottil, T.; Nikghadam-Hojjati, S.; Barata, J. Self-Organization in Smart Manufacturing—Background, Systematic Review, Challenges and Outlook. *IEEE Access* **2023**, *11*, 10107–10136. <https://doi.org/10.1109/ACCESS.2023.3240433>.
2. Müller, M.; Müller, T.; Ashtari Talkhestani, B.; Marks, P.; Jazdi, N.; Weyrich, M. Industrial autonomous systems: a survey on definitions, characteristics and abilities. *at - Automatisierungstechnik* **2021**, *69*, 3–13. <https://doi.org/10.1515/auto-2020-0131>.
3. Sakurada, L.; Leita, P. Multi-Agent Systems to Implement Industry 4.0 Components. In Proceedings of the 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), Tampere, Finland, 2020; pp. 21–26. <https://doi.org/10.1109/ICPS48405.2020.9274745>.
4. Burmeister, B. Industrial Application of Agent Systems: Lessons Learned and Future Challenges. In *Multiagent System Technologies*; Braubach, L.; van der Hoek, W.; Petta, P.; Pokahr, A., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2009; Vol. 5774, *Lecture Notes in Computer Science*, pp. 1–3. https://doi.org/10.1007/978-3-642-04143-3_1.
5. Pulikottil, T.; Estrada-Jimenez, L.A.; Ur Rehman, H.; Mo, F.; Nikghadam-Hojjati, S.; Barata, J. Agent-based manufacturing — review and expert evaluation. *The International Journal of Advanced Manufacturing Technology* **2023**, *127*, 2151–2180. <https://doi.org/10.1007/s00170-023-11517-8>.
6. Sidorenko, A.; Motsch, W.; van Bekkum, M.; Nikolakis, N.; Alexopoulos, K.; Wagner, A. The MAS4AI framework for human-centered agile and smart manufacturing. *Frontiers in artificial intelligence* **2023**, *6*, 1241522. <https://doi.org/10.3389/frai.2023.1241522>.
7. Ferreira, J.; Ribeiro, L.; Neves, P.; Akillioglu, H.; Onori, M.; Barata, J. Visualization tool to support multi-agent mechatronic based systems. In Proceedings of the IECON 2012, Piscataway, NJ, 2012; pp. 4372–4377. <https://doi.org/10.1109/IECON.2012.6389185>.
8. Dorri, A.; Kanhere, S.S.; Jurdak, R. Multi-Agent Systems: A Survey. *IEEE Access* **2018**, *6*, 28573–28593. <https://doi.org/10.1109/ACCESS.2018.2831228>.
9. Ahmad, H.F. Multi-agent systems: overview of a new paradigm for distributed systems. In Proceedings of the Proceedings / 7th IEEE International Symposium on High Assurance Systems Engineering, HASE 2002, Los Alamitos, Calif., 2002; pp. 101–107. <https://doi.org/10.1109/HASE.2002.1173110>.
10. Antsaklis, P. Autonomy and metrics of autonomy. *Annual Reviews in Control* **2020**, *49*, 15–26. <https://doi.org/10.1016/j.arcontrol.2020.05.001>.
11. Chen, H.; Wen, Y.; Zhu, M.; Huang, Y.; Xiao, C.; Wei, T.; Hahn, A. From Automation System to Autonomous System: An Architecture Perspective. *Journal of Marine Science and Engineering* **2021**, *9*, 645. <https://doi.org/10.3390/jmse9060645>.
12. Chen, J.; Abbod, M.; Shieh, J.S. Integrations between Autonomous Systems and Modern Computing Techniques: A Mini Review. *Sensors (Basel, Switzerland)* **2019**, *19*. <https://doi.org/10.3390/s19183897>.
13. Mo, F.; Monetti, F.M.; Torayev, A.; Rehman, H.U.; Mulet Alberola, J.A.; Rea Minango, N.; Nguyen, H.N.; Maffei, A.; Chaplin, J.C. A maturity model for the autonomy of manufacturing systems. *The International Journal of Advanced Manufacturing Technology* **2023**, *126*, 405–428. <https://doi.org/10.1007/s00170-023-10910-7>.
14. Hrabia, C.E.; Masuch, N.; Albayrak, S. A Metrics Framework for Quantifying Autonomy in Complex Systems. In *Multiagent System Technologies*; Müller, J.P.; Ketter, W.; Kaminka, G.; Wagner, G.; Bulling, N., Eds.; Springer International Publishing: Cham, 2015; Vol. 9433, *Lecture Notes in Computer Science*, pp. 22–41. https://doi.org/10.1007/978-3-319-27343-3_2.
15. Pinto, A. An Open and Modular Architecture for Autonomous and Intelligent Systems. In Proceedings of the 2019 IEEE International Conference on Embedded Software and Systems (ICESS), Las Vegas, NV, USA, 2019; pp. 1–8. <https://doi.org/10.1109/ICESS.2019.8782512>.
16. Vagia, M.; Transeth, A.A.; Fjordingen, S.A. A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed? *Applied ergonomics* **2016**, *53 Pt A*, 190–202. <https://doi.org/10.1016/j.apergo.2015.09.013>.
17. Gamer, T.; Hoernicke, M.; Kloepper, B.; Bauer, R.; Isaksson, A.J. The Autonomous Industrial Plant -Future of Process Engineering, Operations and Maintenance. *IFAC-PapersOnLine* **2019**, *52*, 454–460. <https://doi.org/10.1016/j.ifacol.2019.06.104>.
18. Antsaklis, P.J.; Rahnama, A. Control and Machine Intelligence for System Autonomy. *Journal of Intelligent & Robotic Systems* **2018**, *91*, 23–34. <https://doi.org/10.1007/s10846-018-0832-6>.
19. Seitz, M.; Gehlhoff, F.; Cruz Salazar, L.A.; Fay, A.; Vogel-Heuser, B. Automation platform independent multi-agent system for robust networks of production resources in industry 4.0. *Journal of Intelligent Manufacturing* **2021**, *32*, 2023–2041. <https://doi.org/10.1007/s10845-021-01759-2>.
20. Falco, M.; Robiolo, G. A Systematic Literature Review in Multi-Agent Systems: Patterns and Trends. In Proceedings of the 2019 XLV Latin American Computing Conference, Piscataway, NJ, 2019; pp. 1–10. <https://doi.org/10.1109/CLEI47609.2019.235098>.
21. Grieves, M.; Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary perspectives on complex systems*; Kahlen, F.J.; Flumerfelt, S.; Alves, A.C., Eds.; Springer: Cham, 2016; pp. 85–113. https://doi.org/10.1007/978-3-319-38756-7_4.

22. Pretel, E.; Moya, A.; Navarro, E.; López-Jaquero, V.; González, P. Analysing the synergies between Multi-agent Systems and Digital Twins: A systematic literature review. *Information and Software Technology* **2024**, *174*, 107503. <https://doi.org/10.1016/j.infsof.2024.107503>.
23. Kritzinger, W.; Karner, M.; Traar, G.; Henjes, J.; Sihn, W. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* **2018**, *51*, 1016–1022. <https://doi.org/10.1016/j.ifacol.2018.08.474>.
24. Zheng, X.; Psarommatis, F.; Petrali, P.; Turrin, C.; Lu, J.; Kiritsis, D. A Quality-Oriented Digital Twin Modelling Method for Manufacturing Processes Based on A Multi-Agent Architecture. *Procedia Manufacturing* **2020**, *51*, 309–315. <https://doi.org/10.1016/j.promfg.2020.10.044>.
25. Sakurada, L.; de La Prieta, F.; Leitao, P. Digitization of Industrial Environments Through an Industry 4.0 Compliant Approach. In Proceedings of the IECON 2023 - 49th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 2023; pp. 1–6. <https://doi.org/10.1109/IECON51785.2023.10312428>.
26. Ai, L.; Ziehl, P. Advances in Digital Twin Technology in Industry: A Review of Applications, Challenges, and Standardization. *Journal of Intelligent Construction* **2025**, *3*, 1–19. <https://doi.org/10.26599/JIC.2025.9180083>.
27. Mariani, S.; Picone, M.; Ricci, A. About Digital Twins, Agents, and Multiagent Systems: A Cross-Fertilisation Journey. In *Autonomous Agents and Multiagent Systems. Best and Visionary Papers*; Melo, F.S.; Fang, F., Eds.; Springer International Publishing and Imprint Springer: Cham, 2022; Vol. 13441, *Lecture Notes in Artificial Intelligence*, pp. 114–129. https://doi.org/10.1007/978-3-031-20179-0_8.
28. Ricci, A.; Croatti, A.; Mariani, S.; Montagna, S.; Picone, M. Web of Digital Twins. *ACM Transactions on Internet Technology* **2022**, *22*, 1–30. <https://doi.org/10.1145/3507909>.
29. Mariani, S.; Picone, M.; Ricci, A. Towards Developing Digital Twin Enabled Multi-Agent Systems. In *Engineering Multi-Agent Systems*; Ciorrea, A.; Dastani, M.; Luo, J., Eds.; Springer Nature Switzerland and Imprint Springer: Cham, 2023; Vol. 14378, *Lecture Notes in Artificial Intelligence*, pp. 178–187. https://doi.org/10.1007/978-3-031-48539-8_12.
30. Cruz Salazar, L.A.; Vogel-Heuser, B. A CPPS-architecture and workflow for bringing agent-based technologies as a form of artificial intelligence into practice. *at - Automatisierungstechnik* **2022**, *70*, 580–598. <https://doi.org/10.1515/auto-2022-0008>.
31. Lorente, Q.; Villeneuve, E.; Merlo, C.; Boy, G.A.; Thermy, F. Development of a digital twin for collaborative decision-making, based on a multi-agent system: application to prescriptive maintenance. *INCOSE International Symposium* **2022**, *32*, 109–117. <https://doi.org/10.1002/iis2.12875>.
32. Hevner, J.; March, S.; Park, S.; Ram, S. Design Science in Information Systems Research. *MIS Quarterly* **2004**, *28*, 75. <https://doi.org/10.2307/25148625>.
33. Kuechler, W.; Vaishnavi, V. A Framework for Theory Development in Design Science Research: Multiple Perspectives. *Journal of the Association for Information Systems* **2012**, *13*, 395–423. <https://doi.org/10.17705/1jais.00300>.
34. Zhang, J.; Ellwein, C.; Heithoff, M.; Michael, J.; Wortmann, A. Digital twin and the asset administration shell. *Software and Systems Modeling* **2025**, *24*, 771–793. <https://doi.org/10.1007/s10270-024-01255-0>.
35. Barnstedt, E.; Bedenbender, H.; Berres, B.; Billmann, M.; Boss, B.; Braunisch, N.; Braunmandl, A.; Clauer, E.; Diedrich, C.; Flubacher, B.; et al. Details of the Asset Administration Shell - Part 1: The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC02), 2022.
36. Bayha, A.; Bock, J.; Boss, B.; Diedrich, C.; Malakuti, S. Describing Capabilities of Industrie 4.0 Components, 2020.
37. Miadowicz, I.; Quinto, D.M.; Pitz-Paal, R.; Felderer, M. An action research study on the digital transformation of concentrated solar thermal plants. *Solar Energy Advances* **2025**, *5*, 100102. <https://doi.org/10.1016/j.seja.2025.100102>.
38. Deutsches Zentrum für Luft- und Raumfahrt. Solar Towers Jülich, 2025.
39. Zhang, H.L.; Baeyens, J.; Degre, J.; Cacères, G. Concentrated solar power plants: Review and design methodology. *Renewable and Sustainable Energy Reviews* **2013**, *22*, 466–481. <https://doi.org/10.1016/j.rser.2013.01.032>.
40. Schmidt, C.; Volz, F.; Stojanovic, L.; Sutschet, G. Increasing Interoperability between Digital Twin Standards and Specifications: Transformation of DTDL to AAS. *Sensors (Basel, Switzerland)* **2023**, *23*. <https://doi.org/10.3390/s23187742>.
41. Jacoby, M.; Usländer, T. Digital Twin and Internet of Things—Current Standards Landscape. *Applied Sciences* **2020**, *10*, 6519. <https://doi.org/10.3390/app10186519>.
42. Es-haghi, M.S.; Anitescu, C.; Rabczuk, T. Methods for enabling real-time analysis in digital twins: A literature review. *Computers & Structures* **2024**, *297*, 107342. <https://doi.org/10.1016/j.compstruc.2024.107342>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.