When Multi-Agent Systems Meet Digital Twin

Pasquale Rubino*, Sirine Bouguettaya, Vincenzo Barbuto, Claudio Savaglio and Giancarlo Fortino

Dept. of Computer Science, Modeling, Electronics and Systems Engineering (DIMES), Università della Calabria, 87036 Rende, Italy

Abstract

With the rapid maturation of Digital Twin (DT) technologies, physical entities can be enhanced through advanced computational methods applied to their virtual counterparts. A typical DT consists of three core elements: a physical asset, its digital replica, and data connections linking the two. In contrast, the Multi-Agent Systems (MAS) paradigm introduces autonomous entities capable of perceiving, reasoning, and interacting within dynamic environments, thus providing a natural foundation for the development of intelligent and adaptive DT ecosystems. This paper explores the intersection between MAS and DT, with a primary focus on how intelligent agents can transform DTs from reactive entities into proactive and autonomous systems. To this end, we conduct a systematic state of the art analysis following the PRISMA methodology, aiming to identify existing methodologies, architectural designs, and application domains where MAS is integrated with DT technologies. The study is guided by a set of research questions that examine the extent to which MAS contribute to enhancing DT capabilities, as well as the current limitations and research gaps.

Keywords

Digital Twin, Multi-Agent System, Intelligent Agents

1. Introduction

The rapid evolution of cyber–physical systems has accelerated the adoption of Digital Twins (DTs) as a key enabling technology for modeling, monitoring, and optimizing complex physical environments. A DT is defined as a dynamic virtual representation of a physical system that remains continuously aligned with its real world counterpart through data synchronization, enabling real time monitoring, prediction, optimization, control, and informed decision making [1]. This tight integration between the physical and digital worlds allows DTs to support advanced analytics and improve system performance across a wide range of application domains. Beyond this general definition, DTs are characterized by several fundamental properties. These include *representativeness*, which ensures that the digital model captures the essential features of the physical asset; *contextualization*, which incorporates the operational environment of the system; *reflection*, which guarantees that the digital counterpart accurately mirrors the behavior of the physical entity; and *entanglement*, which refers to the strong and continuous connection between the physical and virtual worlds [1]. Despite these advances, most existing DT implementations remain largely reactive, focusing on data visualization, monitoring, and offline analytics. Such approaches often lack the capability to autonomously adapt to dynamic environments or to coordinate complex interactions among multiple system's components [2, 3]. As modern cyber–physical systems grow in scale and complexity, there is an increasing need for more intelligent, decentralized, and adaptive DT ecosystems.

Building on this perspective, Multi-Agent Systems (MAS) emerges as a natural paradigm. An MAS is traditionally defined as a collection of autonomous agents that interact within a shared environment, where coordination, communication, and sometimes competition are required to solve problems that exceed the capabilities of any single agent [4]. By design, MAS support key properties, these characteristics align closely with the requirements of next generation DT, suggesting a natural convergence between

WOA 2026

*Corresponding author.

pasquale.rubino@dimes.unical.it (P. Rubino); sirine.bouguettaya@dimes.unical.it (S. Bouguettaya); vincenzo.barbuto@dimes.unical.it (V. Barbuto); csavaglio@dimes.unical.it (C. Savaglio); giancarlo.fortino@unical.it (G. Fortino)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-10-30

CEUR-WS.org/Vol-4307/paper41.pdf

the two paradigms. Recent research has begun to explore the integration of MAS into DT environments, enabling the transition from passive digital replicas to agent enabled, proactive ecosystems [5]. In such systems, agents can represent physical assets, digital entities, or decision making components, and can collaborate to perform real time monitoring, control, and optimization.

This article aims to provide a comprehensive and insightful survey of the intersection between MAS and DTs, covering their theoretical foundations, architectural frameworks, enabling technologies, application domains, and associated challenges. Indeed, the goal of this survey is twofold: *(i)* to systematically explore the state of the art, following the PRISMA guidelines [6], and to provide a structured framework for analyzing existing contributions that integrate MAS into DT systems from different perspectives, including methodologies and architectural approaches; and *(ii)* to identify and discuss key research gaps, limitations, and practical challenges related to the use of intelligent agents to enhance DT capabilities, as well as to outline potential directions toward more autonomous and agent enabled DT ecosystems.

The paper is organized as follows: Section 2 presents the background on DT and MAS. Section 3 describes the research methodology and PRISMA based selection process. Section 4 synthesizes the collected literature and answers the research questions, and Section 5 concludes the paper.

2. Background

2.1. Digital Twin fundamentals and recent developments

DTs have emerged as a transformative paradigm for modeling and managing physical systems, effectively bridging the physical and digital domains. Although no single, universally accepted definition exists [7], the concept was first introduced by Michael Grieves in 2003 within the context of Product Lifecycle Management [8]. In this original formulation, a DT consists of three core elements: the physical entity, its virtual representation, and the data connections linking the two. This foundational view was later extended by NASA and the U.S. Air Force, which framed DTs as integrated, multi-physics, multi-scale, and probabilistic simulations of as-built systems [9], emphasizing high fidelity modeling and lifecycle prediction. With the advent of IoT technologies enabling pervasive sensing, real-time data exchange, and actuation, DTs have evolved into dynamic virtual models that mirror physical systems continuously, supporting monitoring, prediction, optimization, and decision making [7, 10]. This bidirectional coupling fundamentally distinguishes a DT from related paradigms. In contrast to a Digital Shadow, which supports only a unidirectional flow from the physical entity to its digital representation and is therefore limited to passive observation, a DT closes the loop by enabling feedback actions that directly influence the physical system. Similarly, unlike conventional offline simulations that operate on static or predefined inputs, a DT remains continuously synchronized with the evolving state of a specific physical instance through real-time data exchange. Building on this foundations, Minerva et al. [7] provide a structured characterization of DTs based on two categories of properties. Fundamental properties (representativeness, contextualization, reflection, and entanglement) ensure that the DT accurately captures the relevant features, context, behavior, and coupling with its physical counterpart. Complementing these, optional properties such as composability, augmentation, and predictability extend DT capabilities, particularly in complex Cyber-Physical Systems (CPS), by enhancing analytical power and decision support.

Despite these advances, traditional DT implementations exhibit several important limitations [11, 3, 5]. They are typically passive systems, lacking autonomous or proactive behavior, and thus remain dependent on human intervention for decision-making and control. Their ability to adapt in real time to dynamic environmental conditions is often limited, reducing responsiveness to unexpected events. Moreover, integration with control loops is frequently weak, requiring external supervision to enact changes. Finally, DTs generally lack social capabilities, such as interaction and coordination with other DTs, which constrains their effectiveness in distributed and multi-actor CPS environments. To address these shortcomings, recent research has increasingly explored the integration of Artificial Intelligence (AI) into DT architectures. Following the evolutionary perspective proposed in [12],

DTs can be viewed as progressing through successive maturity stages that progressively enhance key properties such as entanglement and reflection. At the foundational level, the *Predictive DT* augments the digital replica with statistical and machine learning techniques to anticipate future states. The *Reactive DT* introduces rule-based or learned responses to environmental changes, enabling limited autonomous decision-making. At the highest level of maturity, the *Proactive* (or *Cognitive*) *DT* incorporates reasoning, situational awareness, and advanced AI to autonomously interpret context, evaluate alternative actions, and pursue system-level objectives. Complementing this autonomy-oriented progression, the *Opportunistic DT* [13] focuses on adaptability by leveraging synthetic sensing to dynamically expand the observable data space without requiring additional physical instrumentation.

However, while these AI-enhanced DTs significantly improve analytical and decision-making capabilities at the individual level, they do not fully resolve the challenges that arise in complex, distributed ecosystems. As highlighted in recent studies [5], even intelligent DTs largely remain high-fidelity yet reactive entities: they provide insights and recommendations, but actions are still predominantly initiated by humans or external controllers. Furthermore, their limited sociability, (the ability to interact, coordinate, and collaborate with other DTs or autonomous agents) continues to hinder their effectiveness in multi-stakeholder and large-scale cyber-physical environments.

2.2. Intelligent Agents and Multi-Agent Systems for cyber-physical environments

Modern CPS are composed of heterogeneous physical entities (machines, sensors, vehicles, infrastructure components, etc.) endowed with sensing, actuation, processing, and communication capabilities. Despite their constrained resources, these entities are expected to autonomously operate in their local environment, collaborate to deliver services beyond the capability of any single device, and distribute their functionality in a bottom-up fashion across the entire system [14]. Meeting these requirements poses significant development challenges that demand comprehensive support from closely related fields, including distributed systems, AI, and pervasive computing [15].

In this context, the agent-based paradigm offers a natural modeling foundation. A software agent is an autonomous computational entity *situated* in an environment, capable of perceiving its state, reasoning about its goals, and acting upon the environment accordingly [4]. Agents are characterized by key properties such as autonomy, reactivity, proactiveness, and social ability, which collectively enable independent operation, timely response to environmental changes, goal-directed behavior, and coordination with other entities [4]. When multiple agents coexist and interact within a shared environment, they form a MAS [4]. MAS have been acknowledged as a particularly effective paradigm for addressing the challenges of complex CPS [15, 16]. The strong conceptual alignment between software agents and physical entities is well established: the agent model naturally captures the autonomy, proactiveness, and situatedness that characterize these entities, while MAS architectures provide the decentralized coordination, interoperability, and cognitive capabilities that centralized or client/server approaches, which have often led to application-specific, unscalable, and isolated system silos, struggle to deliver at scale [14, 15]. When individual agents are organized into a MAS, the system gains cooperative and social capabilities that mirror the collective behavior expected from distributed physical entities: agents can share information, negotiate resource usage, and jointly deliver complex services that no single entity could accomplish in isolation [14]. This cooperative dimension is essential for modeling systems where physical entities must coordinate across organizational and technological boundaries, adapting their collective behavior to dynamic and unpredictable operating conditions. Furthermore, unlike object-, service-, or component-oriented paradigms, MAS enable modeling both individual entities and entire distributed systems at different levels of granularity and in a technology-agnostic manner, while agent-based simulation supports the validation of multiple design choices, including coordination strategies and interaction protocols, before physical deployment [15]. MAS is also prone to be beneficially integrated with other complementary programming paradigms, including stream, event, and macro-programming approaches, to further address scale-independence, adaptiveness, and situatedness in complex distributed systems such as CPS and IoT [16].

The ability of agents to act as autonomous software surrogates for physical assets, combined with

the MAS capacity for decentralized coordination and adaptive behavior, suggests a strong alignment with the requirements of DT ecosystems. As discussed in the previous subsection, conventional DTs remain largely passive and lack the autonomy, adaptiveness, and coordination capabilities needed to operate effectively in complex and dynamic environments. MAS offer a pathway to address these limitations by enabling a form of virtualization that closely mirrors the structure and dynamics of the real world: each physical asset operating in a given environment is represented as an autonomous agent in the virtual space, and the interactions among these agents (i.e., coordination, negotiation, information sharing, etc.) reproduce the patterns of interaction that occur among real entities in their physical environment [17, 18]. By organizing these agents into coordinated multi-agent architectures, the resulting systems can support the transition from passive digital replicas to intelligent, distributed, and self-adaptive DT ecosystems, a convergence that recent research has increasingly begun to explore and that the remainder of this paper systematically investigates.

3. Research Methodology

3.1. Objectives

This paper explores the intersection between MAS and DT, with a primary focus on how Intelligent Agents enhance DT capabilities. By shifting the perspective from passive digital replicas to agent enabled ecosystems, the study examines how multi-agent paradigms introduce autonomy, coordination, and adaptive behavior into DT environments. The core research questions guiding this study are:

- **RQ1**: What is the contribution of MAS methods to the development of DT ecosystems?
- **RQ2**: How are DT implemented as MAS, and which established agent theory standards govern their design?
- **RQ3**: What are the resulting benefits, limitations, and open research challenges when modeling DT as MAS?

3.2. Search Strategy

An exhaustive search of the articles regarding MAS and DT was performed in February 2026 across the following digital libraries: Scopus, IEEEXplore, and ScienceDirect, we employed a structured, keyword-driven search strategy. The queries were organized around two main conceptual clusters: (i) Multi-Agent paradigms (e.g., MAS, agent-based systems, intelligent agents) and (ii) DT technologies (e.g., digital twin, virtual twin, cyber-physical twin). Considering such key terms and their synonyms, the following search string was identified :

$$\textit{(”Multi-Agent System” OR ”Intelligent Agents”) AND Digital Twin} \quad (1)$$

3.3. Eligibility Criteria

To ensure a high quality analysis of the integration of MAS into DT, explicit inclusion and exclusion criteria were defined. The goal of this work is to examine how MAS are integrated into DT environments, how they are applied across different domains, and what challenges and future research directions emerge from embedding MAS within DT ecosystems. The articles were eligible for selection if they met at least one of the following inclusion criteria:

- The work discusses the integration of MAS within DT ecosystem.
- The study proposes architectures or frameworks in which DT are enhanced by intelligent agents.
- The paper presents DT implemented through Multi-Agent approaches.
- The work employs MAS as a central component of the proposed solution.
- The manuscript is a peer-reviewed articles published in journals, conferences, or book chapters.

The articles were excluded from the selection if they met one of the following exclusion criteria:

- The work focuses exclusively on DT without any integration with MAS or it addresses MAS with no explicit connection to DT technologies [19, 20, 21].
- The study superficially integrates Digital Twin or Multi-Agent concepts or mentions them in a different context (e.g., agent-based simulation, intelligent agents, or future work discussions) without forming a substantive part of the proposed architecture, methodology, implementation, or evaluation [22, 23].
- The manuscript is not peer-reviewed, e.g., a preprint, a technical report, a white paper, an editorial, or an unpublished text.

3.4. Study Selection

The study selection process followed PRISMA 2020 guidelines [6], as shown in figure 1.The initial search across the selected digital libraries (Scopus, IEEE Xplore, and ScienceDirect) using the defined search strings retrieved a total of 1396 articles (596 from Scopus, 668 from IEEE Xplore, and 132 from ScienceDirect). To ensure the relevance and quality of the dataset, several preliminary filtering criteria were applied, including the removal of duplicate records. This initial filtering step resulted in the removal of 372 duplicate records, leaving 1024 publications. In the first screening phase, the remaining papers were evaluated based on their title, abstract, and keywords according to the predefined inclusion and exclusion criteria. Additional exclusion criteria included: (i) non-peer-reviewed publications such as editorials, posters, short papers, and non-technical reports; (ii) papers not written in English; and (iii) studies without full-text availability. Furthermore, studies were excluded if they did not explicitly address the integration of MAS and DT, if MAS or DT were only mentioned superficially, or if no technical architecture or framework was presented. This phase resulted in the exclusion of 761 papers, leaving 263 studies for further analysis. Subsequently, a full-text eligibility assessment was conducted to evaluate the technical contribution and relevance of the remaining papers. During this stage, studies were excluded if they did not present MAS as a core component of the DT architecture, lacked implementation details, or did not demonstrate meaningful interaction between MAS and DT concepts. As a result, 240 papers were excluded, leading to a shortlist of 23 studies. In the final selection phase, a quality assessment was conducted to identify the most relevant and technically rigorous contributions. The selected studies were evaluated based on criteria including clarity of the MAS architecture, level of DT integration, technical depth, and contribution to MAS and DT ecosystem design. Papers that lacked sufficient technical detail or presented only conceptual discussions without architectural validation were excluded. This process resulted in the removal of 8 additional studies, leading to a final selection of 15 primary studies included in the detailed analysis.

4. State of The ART

The intersection between MAS and DT has recently attracted growing research interest due to its potential to enhance system intelligence, adaptability, and autonomy. To systematically organize our survey, the reviewed studies are categorized into three main groups, aligned with the research questions: subsection 4.1 the contribution of MAS methods to the development of DT ecosystems (**RQ1**), subsection 4.2 the implementation of DTs as MAS and the agent-based theories and standards guiding their design (**RQ2**), and subsection 4.3 the resulting benefits, limitations, and open research challenges associated with modeling DTs as MAS (**RQ3**).

4.1. Multi-Agent Approaches for Digital Twins

Referring to RQ1, the analysis of the selected studies shows that MAS contribute to DT ecosystems through several recurring architectural roles. Across the reviewed works, four principal functions stand out (Table 1), describing how MAS facilitate the creation of intelligent and adaptive DT environments. *(i)* **agent based representations of physical assets**, *(ii)* **coordination and orchestration of DT components**, *(iii)* **real time monitoring and data processing**, and *(iv)* **decision support and system**

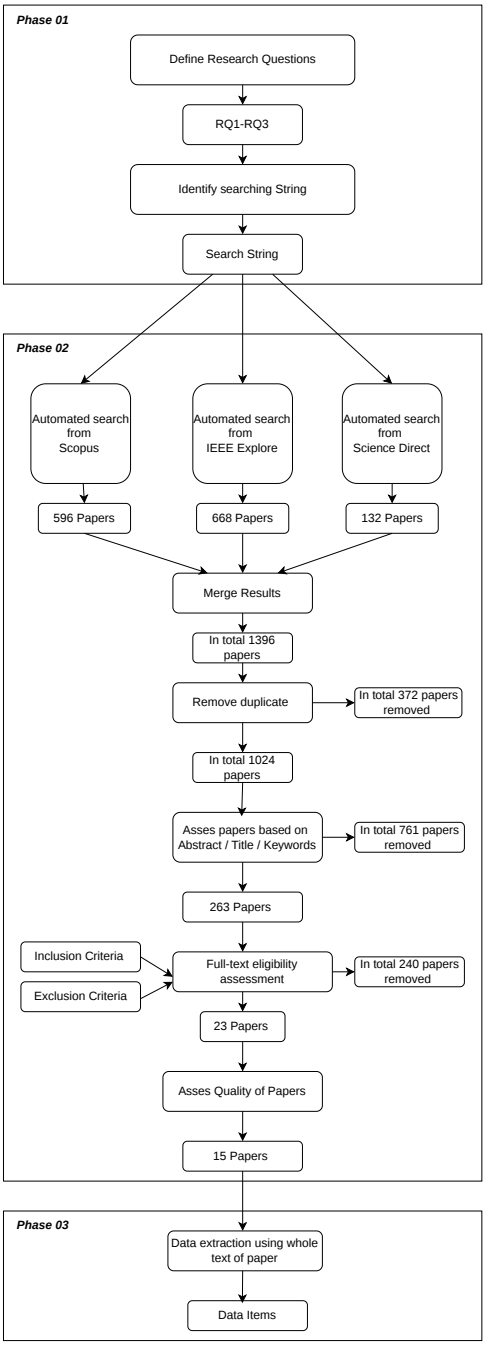

Figure 1: PRISMA-Based Study Selection Process

optimization. First, MAS allows physical system elements to be modeled as **autonomous** agents that act as software surrogates for real world entities such as machines, sensors, vehicles, infrastructure components, or human stakeholders, thus enabling modular and distributed representations of complex systems [18, 24, 17]. Second, MAS offer **coordination** and **orchestration** capabilities via agent communication and negotiation protocols, enabling decentralized management of DT components and reducing the reliance on centralized control mechanisms [25, 18, 24]. Third, MAS enables **real**

time monitoring and data processing by allowing agents to continuously handle IoT and operational data streams and sustain alignment between physical and digital system states [26, 27, 17]. Fourth, MAS supports adaptive **decision making** and **system optimization** through predictive analysis and collaborative reasoning among agents, allowing DT systems toward more intelligent decision support environments [27, 18, 28]. Beyond these functional roles, it is important to understand the underlying MAS properties that enable such capabilities.These functions stem from core MAS properties that shape how agents operate within DT ecosystems. MAS and their agents are defined by autonomy and self-management capabilities [29, 27, 18, 17, 30, 31, 32], which allow both physical and digital DT elements to function with limited human oversight and to adjust their behavior in response to changing environmental conditions [29, 18]. Moreover, MAS enhance **interoperability** and **communication** via standardized interaction patterns and ontology based system descriptions, which ease integration with APIs, external tools, and heterogeneous components[25, 29, 18, 28, 31, 32]. In addition, MAS possess qualities such as **pro-activeness**, **adaptability**, and **collaborative behavior**, enabling agents to cope with uncertainty and dynamically modify their actions based on real-time information and overarching system objectives [25, 29, 30, 28, 31, 32]. As a result of these functional roles and enabling properties, MAS also generates significant impacts on the performance and reliability of DT ecosystems. These impacts include improved **robustness**, **fault tolerance**, **resilience**, **flexibility**, and **reliability** [25, 29, 18, 17, 30, 28, 33, 32], enabling DT systems to handle failures and uncertain conditions without compromising overall system stability. The modular nature of MAS also promotes scalability and supports hierarchical system decomposition [25, 18, 17, 28], allowing DTs to represent systems at multiple levels of granularity, from individual components to holistic system-level views [29]. Moreover, MAS improves **decision making** processes through distributed intelligence, collaborative reasoning, and advanced orchestration mechanisms such as BDI-based approaches, enabling self adjustment, replanning, and adaptive optimization of DT operations [27, 28, 33, 32]. Overall, these findings indicate that MAS acts as an enabling intelligence layer that transforms DTs from passive digital representations into distributed, adaptive, and autonomous cyber-physical ecosystems capable of continuous monitoring, coordination, and intelligent decision-making.

Table 1
Contributions of MAS to Digital Twin Ecosystems

MAS architectural role	MAS design principle	DT Capability	System-level impact	Ref
Agent based representations of physical assets	Autonomy, modularity	Digital representation of physical entities and modular system modeling	Improved modularity, scalability, and distributed DT modeling	[18, 24, 17]
Distributed coordination and orchestration	Social ability, cooperation, Communication,Negotiation	Distributed management of DT components and agent collaboration	Reduced reliance on centralized control, improved interoperability and system efficiency	[18, 24, 25]
Real time monitoring and data processing	Reactivity, Adaptability	Continuous synchronization between physical and digital systems	Improved synchronization between physical and digital environments and enhanced situational awareness	[17, 26, 27]
Intelligent decision support and optimization	Proactiveness, Collaboration, Reasoning	Predictive analysis and adaptive system control	Improved system optimization, adaptive behavior, and intelligent DT decision support	[18, 27, 28]

4.2. Digital Twin implemented as Multi-Agent Systems

Referring to RQ2, the analysis of the selected studies shows that DT are increasingly implemented using MAS architectures, where system components are represented as interacting agents operating within distributed cyber-physical environments. This architectural approach allows complex DT ecosystems to be decomposed into autonomous and interacting entities capable of coordination, monitoring, and adaptive decision making (Table 2). In practice, such implementations rely on distributing intelligent functionalities across agents and DT components, an aspect that has been formally conceptualized in [34], where the integration of MAS and DT is guided by design principles such as *specificity*, *scope*, and *timing*.

To formally model the agent based DT systems, several studies use **Agent-Based Modeling (ABM)** as their main conceptual framework. ABM is a *computational modeling methodology* in which complex systems are modeled as ensembles of **autonomous**, **interacting agents** embedded in a common environment, with overall system behavior arising from their local interactions [17]. The central construct in this paradigm is the *intelligent agent*, defined as an autonomous computational unit capable of making decisions, interacting with other entities, and adapting according to its internal goals and environmental conditions [30]. Such agents commonly possess characteristics including autonomy, social capability, reactivity, proactiveness, and adaptability, which makes them especially well suited for capturing the dynamics of complex CPS [30]. ABM enhances how DT are represented in three key ways. First, it offers a more realistic representation of physical systems by modeling diverse entities as autonomous software agents that interact with each other. Second, it facilitates distributed and scalable models by naturally mirroring the decentralized organization of CPS. Third, it enables the examination of emergent behaviors arising from agent interactions, which are often challenging to capture with conventional modeling methods.Furthermore, ABM is especially well suited for DT implementations because agents operate autonomously and can incorporate heterogeneous data sources, including IoT sensor networks, thereby improving scalability and robustness [17]. Moreover, the use of ontologies and knowledge bases strengthens semantic consistency and supports more effective decision making [24], while the integration of real time IoT data ensures ongoing synchronization between the physical and virtual environments [30].

Beyond the modeling perspective, the practical deployment of MAS based DT demands the actual implementation of the individual agents that constitute the system. MAS based DT are generally conceived at the system level as ensembles of interacting agents that model both physical and digital entities. After establishing the MAS based DT architecture, attention moves to implementing the individual agents that make up the overall system. At this stage, agents are frequently realized using reactive agent architectures, which, through perception action mechanisms, allow them to respond rapidly to changes in their environment, an ability that is especially valuable in dynamic DT settings. As an illustration, the MADTwin framework demonstrates how individual DT agents can be built following agent oriented programming principles, where their behavior unfolds in perception decision action cycles to facilitate intelligent system management [17]. Its extension further demonstrates how additional reasoning capabilities can be embedded within individual agents to support adaptive decision making in distributed CPS [28]. At the individual agent implementation level, and consistent with the reactive architectures adopted in frameworks such as MADTwin [17, 28], MAS based DT rely on established agent theory models, including reactive agents for real time responses and cognitive agent models such as the **Belief–Desire–Intention (BDI)** architecture, which enable goal oriented reasoning, planning, and adaptive decision making [28]. From a technological perspective, the implementation of these agent architectures is typically supported by dedicated MAS development platforms and communication standards, reactive architectures frequently serve as the fundamental approach, while more sophisticated cognitive paradigms, such as the BDI architecture, can be integrated to support goal driven reasoning, planning, and adaptive decision making when enhanced intelligence is needed [28]. From an implementation perspective, these reasoning models are typically supported by MAS development platforms and standards such as **JADE** and **BDI** extension **BDI4JADE** for agent deployment and cognitive reasoning support [28], as well as by agent oriented development frameworks like **PADE**,

JADEX, and **JaCaMo**, which provide capabilities for BDI reasoning, agent communication, and multi-agent coordination [32]. Moreover, **IoT communication** mechanisms and **FIPA-ACL** communication standards are widely used to support interoperability and data exchange between physical assets and their virtual representations [28]. However, recent systematic studies indicate that, while MAS offer substantial potential for enabling intelligent DT ecosystems, several MAS features including learning, collaboration, and advanced reasoning are still underutilized in existing implementations, underscoring the need for additional research toward more holistic MAS based DT design methodologies [32].

Table 2
Technologies and Standards for MAS-Based Digital Twin Implementation

Category	Technology	Role in MAS-DT implementation	Ref.
Modeling Paradigm	ABM	Formal modeling and simulation of DT environments as interacting agents	[17, 29]
Agent reasoning models	Reactive Agent	Real time response and environment interaction	[11]
	BDI	Goal oriented reasoning and adaptive decision making	[28]
MAS development platforms	JADE	Agent deployment and lifecycle management	[28, 29]
	BDI4JADE	Integration of cognitive BDI reasoning within MAS platforms	[28]
	JADEX	BDI based agent development and reasoning support	[32]
	JaCaMo	Integration of agents, environment, and organizational models	[32]
	PADE	Distributed MAS development and communication support	[32]
Communication standards	FIPA-ACL	Standardized agent communication protocols	[28]
	IoT communication mechanisms	Data exchange between DT components and agents physical and digital systems	[28]

4.3. Benefits, challenges, and emergent properties

Referring to RQ3, as described before, MAS significantly enhances DT through autonomous agent-based modeling, real-time monitoring, and adaptive decision making, while also introducing several challenges (Fig. 2). These capabilities improve system **scalability**, **flexibility**, **interoperability**, and **resilience**, while reducing reliance on centralized control. Furthermore, MAS supports efficient resource management and continuous system optimization, transforming DT into dynamic, self-adaptive CPS. But despite there being several advantages in using MAS systems to support DT, various challenges remain. First Implementing DT with MAS can be **resource intensive**, so examining the **economic feasibility** and **resource demands** of such implementations is crucial [25],this above all because most agent based and MAS programming languages and frameworks [29, 28, 33] are are complex and difficult to use and deploy. They need much effort, time, and programming skills to utilize, a possible solution to that is proposed in [29], they consider using a low-code development approach and model-driven engineering concepts and techniques to add layers of abstraction, which could simplify and speed up the process of building DT by having multi purpose physical and digital agent models that can be altered and used for new DT implementations. In addition to the design and implementation phase, there are also problems related to testing, in particular The most significant general gaps identified are consistent evaluation and benchmarking, with a lack in measurability and comparability[31]. This

is related also to the fact that most studies are prototypical [31], addressing partial aspects without comprehensive and methodologically sound validation, while future work is mostly superficial and general in scope, with also a lack for standardization and unification [31], especially in context with heterogeneity [31] . This results in a strong focus on architectural prototypes rather than frameworks with comparable performance. Not only is there a bias on digital shadow[32] rather than DT. This leads to a lack of solid theoretical basis[32], In this regard, there is also a need for the research community to join efforts to provide methodologies that facilitate the design of DT using MAS as the development approach. The release of new frameworks customized for guiding and scaffolding both DT design and development using MAS methodologies and the definition of novel theoretical proposals that provide a basis on how to develop DT supported by MAS would also be a step towards the maturity of the DT field. Added to this are the challenges of implementation. Ensuring seamless **interoperability**[25, 31] and **scalability**[31], defining and measuring the **performance** and **effectiveness**[25] between different systems and technologies is a key challenge that needs to be addressed. Furthermore, **achieving** and **maintaining real time data** synchronisation and **accuracy**[25, 31] in dynamic environments presents a significant challenge that must be overcome. In fact, **real time capability** and **determinism** in a production environment remains critical development factors, including agent lifecycle management, monitoring, and error handling. These factors account for fewer reports on closed loop actuation than on advisory or monitoring decision support tools. Linked to this, the challenges of **efficiently** gathering and handling **high frequency data** from physical components[29, 28]. Jittering, delays, and inaccuracies in data may cause undesirable results and damages. In addition, **anomalies**, **inconsistencies**, and the lack of **controllable semantic**[31] mappings in this data can result in undesirable behavior in the physical system, as the DT are incapable of making the correct decisions. To tackle these implications, some suggested solutions aim to integrate dependable mechanisms such as *time series* technologies in agents for handling big data and using complex event processing methods to detect deviations. As specified in [30], from a technical perspective there are also a lack of highly generalizable AI planning algorithms, incomplete industry knowledge graphs, insufficient uniformity in multi paradigm modeling language, closed nature of commercial software ecosystems, lack of inter-organizational trust mechanism. Finally, ethical and privacy concerns[25], including data security and user consent, must be thoroughly examined.

5. Conclusion

DT ecosystems need to be truly supported by pervasive, autonomous, and adaptive intelligence to unleash their full potential in complex cyber-physical environments. AI-driven enhancements have so far helped, progressing from Predictive to Reactive and Proactive DTs, but the latest paradigm to emerge as a natural complement is MAS, whose growing, though still nascent, literature reflects its broad appeal for transforming passive digital replicas into distributed, self-adaptive, and socially capable ecosystems. This survey provided both a quantitative and qualitative analysis of the intersection between MAS and DT, systematically reviewing the accumulated body of knowledge by means of a PRISMA-based methodology that, starting from 1396 candidate papers, led to the identification and in-depth analysis of 15 primary studies.

As a final takeaway, we recognize that MAS contribute to DT environments through four well-defined architectural roles, i.e., agent-based asset representation, distributed coordination, real-time monitoring, and intelligent decision support, grounded in core agent properties such as autonomy, reactivity, proactiveness, and social ability. However, the realization of these capabilities remains largely prototypical, and intelligence has not yet been considered as a built-in feature of DT architectures. As a result, it is still to be fully explored how advanced agent capabilities, including learning, BDI reasoning, and collaborative planning, should be systematically embedded into DT systems to achieve their maximum yield, while further developments are needed in unified design methodologies, standardized benchmarks, interoperability mechanisms, and trust and governance frameworks. In particular, the convergence of MAS and DT, at its full realization, may lead to truly agentic DT ecosystems where

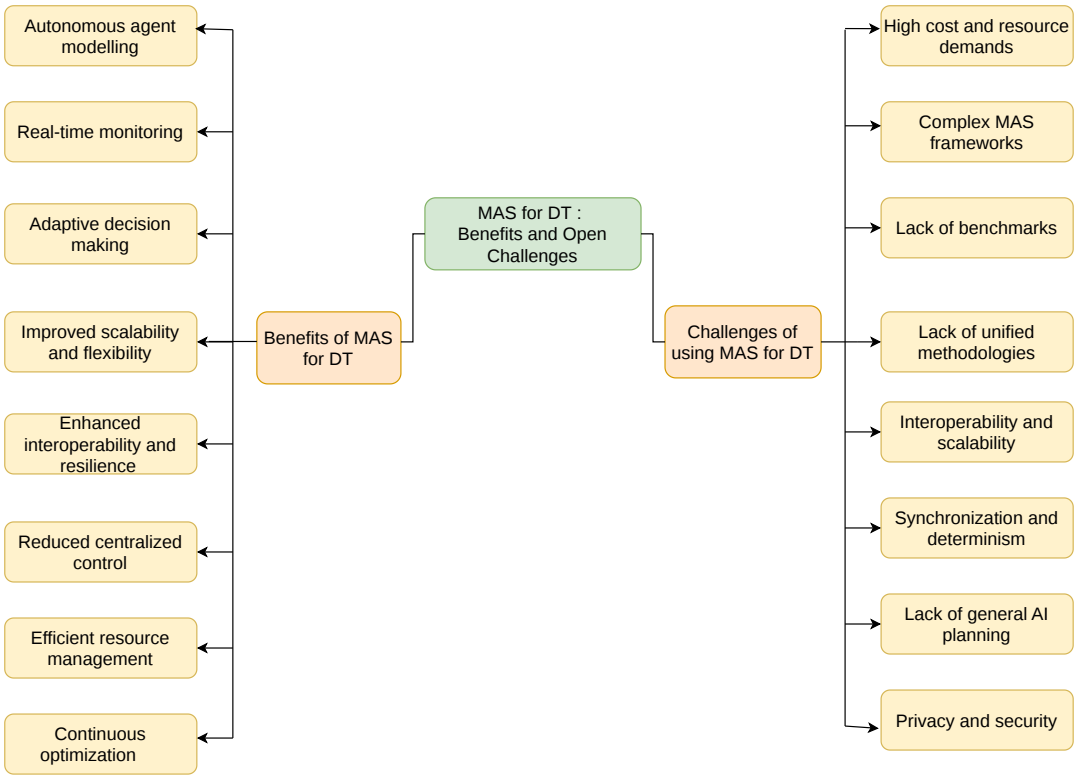

Figure 2: Benefits and Open Research Challenges in MAS-Based Digital Twins.

autonomous agents not only mirror but actively manage, coordinate, and optimize their physical counterparts. From such a perspective, intelligence can be distributed dynamically across the ecosystem, ultimately dissolving the boundaries between passive digital representations and autonomous cyber-physical entities.

To conclude, the identified synergies between MAS and DT paradigms indicate a substantial and largely untapped potential for advancing the next generation of intelligent, autonomous, and socially capable DT ecosystems. This survey is intended to serve as a structured reference to orient future research efforts, support the development of more rigorous and unified methodologies, and contribute to bridging the gap between conceptual proposals and industrially viable MAS-based DT systems.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] R. Minerva, G. M. Lee, N. Crespi, Digital twin in the iot context: A survey on technical features, scenarios, and architectural models, Proceedings of the IEEE 108 (2020) 1785–1824.

[2] L. M. Reinpold, et al., Systematic comparison of software agents and digital twins, Journal of Intelligent Manufacturing 36 (2025) 765–800.

[3] D. Helbing, J. Argota Sánchez-Vaquerizo, Digital twins: Potentials, ethical issues and limitations, in: Handbook on the Politics and Governance of Big Data and Artificial Intelligence, Edward Elgar Publishing, 2023, pp. 64–104.

[4] M. Wooldridge, An Introduction to Multiagent Systems, John Wiley & Sons, 2009.

[5] L. M. Reinpold, L. P. Wagner, F. Gehlhoff, M. Ramonat, M. Kilthau, M. S. Gill, J. T. Reif, V. Henkel, L. Scholz, A. Fay, Systematic comparison of software agents and digital twins: Differences, similarities, and synergies in industrial production, Journal of Intelligent Manufacturing 36 (2025) 765–800.
[6] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, et al., The prisma 2020 statement: an updated guideline for reporting systematic reviews, bmj 372 (2021).
[7] R. Minerva, G. M. Lee, N. Crespi, Digital twin in the iot context: A survey on technical features, scenarios, and architectural models, Proceedings of the IEEE 108 (2020) 1785–1824.
[8] M. Grieves, Digital twin: manufacturing excellence through virtual factory replication, 2022. White paper (original work published 2014).
[9] E. Glaessgen, D. Stargel, The digital twin paradigm for future nasa and us air force vehicles, in: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2012, p. 1818.
[10] D. Jones, C. Snider, A. Nassehi, J. Yon, B. Hicks, Characterising the digital twin: A systematic literature review, CIRP journal of manufacturing science and technology 29 (2020) 36–52.
[11] S. Mihai, M. Yaqoob, D. V. Hung, W. Davis, P. Towakel, M. Raza, M. Karamanoglu, Digital twins: A survey on enabling technologies, challenges, trends and future prospects, IEEE Communications Surveys & Tutorials 24 (2022) 2255–2291.
[12] C. Savaglio, V. Barbuto, F. Mangione, G. Fortino, Generative digital twins: A novel approach in the iot edge-cloud continuum, IEEE Internet of Things Magazine 8 (2024) 42–48.
[13] V. Barbuto, et al., Opportunistic digital twin: an edge intelligence enabler for smart city, ACM (2023).
[14] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, G. Fortino, Agent-based internet of things: State-of-the-art and research challenges, Future Generation Computer Systems 102 (2020) 1038–1053.
[15] G. Fortino, W. Russo, C. Savaglio, Agent-oriented modeling and simulation of iot networks, in: 2016 federated conference on computer science and information systems (FedCSIS), IEEE, 2016, pp. 1449–1452.
[16] R. Casadei, F. Fornari, S. Mariani, C. Savaglio, Programming iot systems: A focused conceptual framework and survey of approaches, Internet of Things 31 (2025) 101548.
[17] T. Clemen, et al., Multi-agent systems and digital twins for smarter cities, in: Proceedings of the 2021 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, ACM, 2021.
[18] Q. Nie, et al., A multi-agent and internet of things framework of digital twin for optimized manufacturing control, International Journal of Computer Integrated Manufacturing 35 (2022) 1205–1226.
[19] G. Cappiello, et al., From digital tools to sustainable change: How change agents enable the twin transition in hospitality, International Journal of Hospitality Management 134 (2026) 104591.
[20] H. Fan, et al., A unified framework for large language model-guided reinforcement learning in digital twin industrial environments, Robotics and Computer-Integrated Manufacturing 99 (2026) 103215.
[21] X. Zhang, et al., Dt-aided resource allocation via generative adversarial imitation learning in complex cloud-edge-end scenarios, Scientific Reports 16 (2026) 7657.
[22] T. Yeung, J. G. Martinez, J. Schlenger, A. Borrmann, R. Sacks, Integrating digital twin and agent-based simulation to support adaptive production system design in building projects, Automation in Construction 180 (2025) 106550. doi:10.1016/j.autcon.2025.106550.
[23] K. J. Prabuchandran, H. K. A. N., S. Bhatnagar, Multi-agent reinforcement learning for traffic signal control, in: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 2014, pp. 1897–1902. doi:10.1109/ITSC.2014.6957986.
[24] L. Xu, et al., Multi-agent digital twinning for collaborative logistics: Framework and implementation, Journal of Industrial Information Integration 45 (2025) 100799.
[25] Y. Kalyani, R. Collier, The role of multi-agents in digital twin implementation: Short survey, ACM

Computing Surveys 57 (2024) 1–15.
[26] X. Zheng, et al., A quality-oriented digital twin modelling method for manufacturing processes based on a multi-agent architecture, in: Procedia Manufacturing, volume 51, Elsevier, 2020, pp. 309–315.
[27] P. Skobelev, et al., Multi-agent approach for developing a digital twin of wheat, in: Proceedings of the 2020 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2020.
[28] H. Marah, M. Challenger, Adaptive hybrid reasoning for agent-based digital twins of distributed multi-robot systems, Simulation 100 (2024) 931–957.
[29] H. Marah, M. Challenger, Madtwin: a framework for multi-agent digital twin development: smart warehouse case study, Annals of Mathematics and Artificial Intelligence 92 (2024) 975–1005.
[30] J. Yu, J. Zhou, J. Fu, From digital twin to digital twin agent, in: Proceedings of the 1st International Symposium on E-CARGO and Applications, IEEE, 2025.
[31] J. Lehmann, et al., Enabling intelligence in digital twins through semantic enrichment and multi-agent systems: A systematic literature review, IEEE Access (2026).
[32] E. Pretel, et al., Analysing the synergies between multi-agent systems and digital twins: A systematic literature review, Information and Software Technology 174 (2024) 107503.
[33] H. Marah, M. Challenger, Intelligent agents and multi agent systems for modeling smart digital twins, in: Engineering Multi-Agent Systems, Springer, 2022.
[34] S. Burattini, S. Mariani, S. Montagna, M. Picone, A. Ricci, Distributing intelligent functionalities in the internet of things with agents and digital twins, Internet of Things 31 (2025) 101560. doi:10.1016/j.iot.2025.101560.