Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

DISTRIBUTED, HIERARCHICAL DIGITAL TWINS: STATE-OF-THE-ART, CHALLENGES AND POTENTIAL SOLUTIONS

Aatu Kunnari[1], and Steffen Strassburger[1]

[1]Group for Information Technology in Production and Logistics, Technische Universität Ilmenau, Ilmenau, GERMANY

ABSTRACT

Digital Twins (DT) provide detailed, dynamic representations of production systems, but integrating multiple DTs into a distributed ecosystem presents fundamental challenges beyond mere model interoperability. DTs encapsulate dynamic behaviors, optimization goals, and time management constraints, making their coordination a complex, unsolved problem. Moreover, DT development faces broader challenges, including but not limited to data consistency, real-time synchronization, and cross-domain integration, that persist at both individual and distributed scales. This paper systematically reviews these challenges, examines how current research addresses them, and explores their implications in distributed, hierarchical DT environments. Finally, it presents preliminary ideas for a structured approach to orchestrating multiple DTs, laying the groundwork for future research on holistic DT management.

1 INTRODUCTION

Digital Twins of production and logistics systems can support real-time decision making through data-driven models. They digitally replicate the processes of and integrate live data from the real system. One typical objective of DTs is fine-tuned control of the real system (Kuehner et al. 2021).

From a simulation-centric perspective, a DT obviously must replicate the dynamic behavior of the real system and should provide a "fast-forward" option to allow the evaluation of control alternatives. Simulation scientists therefore tend to see a simulation model as the core feature of a DT. Other obviously required features include experiment control facilities (for evaluating decision alternatives or enabling simulation-based optimization) and a strategy for feeding real-time data into the DT. Also, a strategy for feeding control decisions back into the real system is needed (Kuehner et al. 2021).

Still, the question of how to build a DT needs to be answered. A possible way to create a DT is the ISO 23247 standard family. This standard offers a framework to build single DTs but lacks components of multi-DT integration (Shao et al. 2023). It also lacks "concrete solutions", functioning as a guideline in decision making, rather than a blueprint. It misses the ability to provide a ready-to-use toolkit to create DTs. ISO 23247 is being adopted in industry, providing a structured framework for DT development. However, it does not define concrete software solutions, leaving implementation details to individual developers. As a result, DT ecosystems remain heterogeneous, making interoperability a persistent challenge.

Since the connection of DTs is a challenge not addressed by current standards, the question arises: How could one build a system that would be able to incorporate multiple DTs in an easy to implement way. Would such a system need a monolithic structure, that encompasses entire DTs inside of it or would a modular plug-and-play system offer a better solution?

To mitigate the need for modular and composable DTs further, let us consider a factory-level DT. It might provide a high-level representation of material flow and production processes but lack detailed insights into individual machine behavior or current operating status. Similarly, a DT of a machine could provide deeper insights into its behavior and its current state, improving the accuracy of process optimizations and predictive analytics (Poechgraber et al. 2023), but they might lack the ability to aim for global optimization goals of a factory. The question arises: Can a coupling of these different types of DTs increase the fidelity and functionality of the virtual depiction of the factory?

These different types of DTs will need to be created by domain experts who specialize in their own areas leading to more complex and heterogenous solutions. As DTs become more complex and interdependent, simple data exchange mechanisms alone become insufficient for their coordination. The different DTs have heterogeneous attributes, functionalities, and software solutions, which raises the next question: How can these diverse DTs be integrated into a modular, interconnected DT environment, and what kind of hierarchical structure emerges from their interconnection?

When multiple DTs work on the same physical entity, they need to communicate and coordinate with each other as to not cause problems within the physical system. Without a structured coordination mechanism, DTs risk conflicting optimization goals, conflicting commands and inconsistent real-time behavior. To address these challenges, a hierarchical structure becomes necessary, leading to the Distributed, Hierarchical Digital Twin (DHDT) paradigm. A DHDT could take three forms: (1) a comprehensive DT that integrates multiple specialized DTs into a cohesive system, (2) a network of interconnected DTs that collectively represent a complex production process, or (3) a mix of the first two approaches. In all cases, a hierarchical structure emerges where a supervisory DT governs optimization goals while subordinate DTs contribute domain-specific insights. DHDTs offer a more granular representation of production processes, leading to potentially better results.

However, the integration of multiple DTs into a DHDT potentially extends challenges encountered in traditional DT implementations. While existing research has addressed these challenges in the context of standalone DTs, they require renewed examination within the DHDT paradigm. Key challenges include the integration of new components, synchronization of distributed simulation models, communication between digital and physical assets, scalability, and distributed governance. The interplay between multiple DTs introduces complexities in interoperability, real-time coordination, and cross-domain decision-making that are not present in isolated DT environments.

This paper aims to provide a structured review of research on distributed and hierarchical DTs. Specifically, we focus on challenges in the implementation, development, and integration of these DTs. The objective is to bridge the gap between established DT methodologies and the emerging requirements of distributed, hierarchical implementations.

The rest of the paper is structured as follows:

- Section 2 introduces key concepts relevant to DHDTs, including Digital Twins, distributed systems, and hierarchical system architectures.
- Section 3 presents a review of distributed DTs (DDT) and hierarchical DTs (HDT)
- Section 4 describes a preliminary architecture for a DHDT and summarizes the requirements for implementation
- Section 5 concludes the paper and outlines directions for future research.

2 FUNDAMENTALS: DIGITAL TWINS, DISTRIBUTED SYSTEMS AND HIERARCHICAL DIGITAL TWINS

To understand the DHDT paradigm, it is essential to first explain the different concepts involved, starting with DTs, followed by a brief explanation on what contributes to a distributed system and finally what hierarchies mean in the context of DHDTs and this paper.

2.1 Digital Twins

This paper adopts the definition from Kuehner et al. (2021), who conducted a meta-review to find a commonly accepted definition in the field. According to their review, a DT is a virtual representation of a physical entity with a bidirectional data connection, enabling simulation-based decision-making across multiple phases of the entity's lifecycle. This definition is broadly adaptable across various areas of DT development.

DTs are used for optimization, control, decision support, and predictive analysis (Kuehner et al. 2021; Mihai et al. 2022). In industrial applications, Production DTs play a central role by enabling process optimization, resource management, and real-time control (Dietz and Pernul 2020). Their key objectives include reducing costs, minimizing machine idle times, improving energy efficiency, and optimizing material flow and maintenance scheduling, all while maintaining production throughput and system resilience.

Product DTs provide a more granular view by representing individual items within the production system or during production development. These can include machines within a production line (Poechgraber et al. 2023), batteries (Cheng et al. 2021; Fatemeh et al. 2023) or products during their development (Strelets et al. 2020). Machine DTs enable predictive maintenance, ensuring longer lifecycles and reduced downtime (Aivaliotis et al. 2019). Product DTs track items from design to recycling, supporting product development (Li et al. 2021) and lifecycle management (Lehner et al. 2024).

Other types of DTs include Building DTs (Hammar and Stadler 2023; Zhang 2023), Power Grid DTs (Gao et al. 2022), and Supply Chain DTs (Barykin et al. 2020). Although Building DTs are relevant to the production environment, their integration within the DHDT paradigm is conceptual and not the focus of this paper.

Similarly relevant to the production system, the Power Grid DTs and Supply Chain DTs could enable cross-company or cross-industry optimization, but they are constrained by limited control from the manufacturing perspective. However, if the DHDT paradigm is widely adopted, these DTs could be integrated into a larger, interconnected network in the future.

All these different DTs operate on their own specific scale, using independent datasets and pursuing distinct optimization goals. This can lead to conflicting objectives, such as a Production DT aiming to maximize output while a Product DT seeks to extend equipment lifespan. The DHDT paradigm must address these conflicts by structuring DTs into a coordinated hierarchy, enabling cross-domain optimization and holistic system improvements. This paradigm not only aims to enhance individual DT functionalities but also to lay the groundwork for future research into scalable, interconnected digital ecosystems.

2.2 Distributed Systems

Veríssimo (2001) identifies three key attributes that characterize distributed systems: multiple computers, a network based interconnection, and a sharing of system state. Van Steen and Tanenbau (2016) on the other hand defines a distributed system as "… a collection of autonomous computing elements that appears to its users as a single coherent system". With these two approaches in mind, we can analyze DTs and their distributed nature.

DTs are parts of a distributed system. The distributed system consists of the DT itself, sensors inside the factory providing real-time data, and data exchange systems connecting the DT and the physical system. But the DT itself does not have to be a distributed system. Depending on the definition of the DT's system borders it could either be considered a distributed system or not. If the DT uses for example cloud services, to process data or run optimization algorithms it is distributed. On the other hand, if the DT processes all data locally and doesn't utilize distributed resources it is not a distributed system.

Whereas DTs themselves might or might not be distributed systems, a collaborative network of DTs most certainly is. The notion of multiple systems of simulation models, optimization algorithms, and data connections, that run independently from each other, yet observing the same system and its states, and present as a single coherent system, fills the required characteristics for a distributed system.

As DTs become more interconnected, managing their interactions becomes increasingly complex. While individual DTs operate autonomously, collaborative networks of DTs must function as a unified system while preserving local control. This introduces critical challenges such as:

- Coordinating real-time communication across DTs.
- Managing conflicting optimization goals between DT components.
- Ensuring consistency and synchronization across distributed simulation models.
- Managing different time requirements of different models.

These challenges necessitate a structured coordination mechanism, leading to the Distributed, Hierarchical Digital Twin (DHDT) paradigm.

DHDTs need to be able to present to the observer as a single entity, even though many of its capabilities are distributed onto different DTs, which in and of themselves are (possibly) distributed systems and need to be perceived as uniform systems. So rather than calling DHDT a simple distributed system, we should see it as distributed system of systems. They bring all the challenges of a distributed system with them, but amplify them, by adding a layer of necessary coordination on top of them.

2.3 Hierarchical Systems

Hierarchies in DTs are traditionally understood as levels of system (Rayhana et al. 2024). At the highest level, a DT may represent an entire factory; at lower levels, individual machines or specific components like motors are modeled. This form of hierarchy, here referred to as structural hierarchy, helps organize and decompose complex systems.

However, in a distributed, hierarchical digital twin (DHDT) environment, hierarchy involves more than just system granularity. It encompasses behavioral hierarchy: the distribution of decision-making authority, system access, optimization priority, and orchestration rights among interlinked DTs. For example, a factory-level DT might prioritize global production targets, while a machine-level DT focuses on local energy efficiency. When conflicts arise, such as production speed versus energy consumption, the behavioral hierarchy determines which DT's optimization goal takes precedence.

This hierarchy is not necessarily static. It may emerge from factors such as the level of detail in the DT, the engineer's familiarity with underlying algorithms, or the degree of access to a DT developed by another party. The resulting hierarchy shapes simulation orchestration, data-sharing permissions, and the influence of localized DTs on global system behavior. In this sense, behavioral hierarchy acts as a governance mechanism within DHDT ecosystems.

3 DISTRIBUTED, HIERACHICAL DIGITAL TWINS: STATE OF THE ART

The concept of DHDTs remains largely unexplored. Little to no literature on the combined topic of distributed and hierarchical twins was found in Scopus, the ACM Digital Library, Google Scholar, or IEEE Xplore. Therefore, we conducted two separate systematic literature reviews to identify relevant research under alternative terminologies.

3.1 Distributed Digital Twins

A targeted search within major academic databases was performed using key terms such as "distributed digital twins," "federated digital twins," and "multi-agent digital twins." This process yielded 24 potential sources, with 16 ultimately included in the final review. To extract patterns across these publications, an initial concept matrix was developed and later transformed into a morphological box to support a more structured representation of recurring themes. The dimensions of the morphological box were not selected arbitrarily; rather, they emerged inductively during the early analysis phase. Recurring topics such as governance models, architectural types, and optimization objectives were consistently observed across the sources. Their frequency and relevance prompted the structured cross-comparison that underpins Table 1.

The left column of the morphological box shows six different categories that were analyzed in the reviewed literature. Many papers fell into multiple varieties within a single category. The numbers listed in the table indicate how many sources addressed each topic. The categories are:

- Subject of Distribution: Is the DT itself a distributed system or are there multiple distributed DTs, which together form a distributed system?
- Governance Type: How are DDT orchestrated? Is there a central coordinator or some other system?
- Architecture Type: What kind of system architecture was developed?
- Presence of Hierarchies: Do the authors describe hierarchies in any form in their publication?
- Objectives: What are the DDTs intended to do?
- Challenges: What challenges are addressed or acknowledged?

Table 1: Morphological box on distributed Digital Twins review.

Subject of Distribution	One distributed DT (7)		Multiple distributed DTs (11)	
Governance Type	Top-Down (2)	Centralized-Decentralized (2)	Supervised Decentralized (1)	Decentralized (2)
Architecture Type	Distributed (5)	Micro Services / Service Oriented (7)		Monolithic (1)
Presence of Hierarchies	Yes (3)		No (13)	
Objectives	System Optimization / Control (7)	Asset Optimization / Control (11)	Modular Modeling (2)	Shared Learning (5)
Challenges	Data Management (10)	Connectivity (11)	Governance (7)	Resources (11)

While the morphological box offers a structured categorization of recurring topics in DDT literature, cross-category relationships were limited. For example, hierarchies rarely co-occurred with specific governance models or architectural types. Similarly, objectives such as shared learning or asset monitoring appeared independently of system structure. This absence of strong co-occurrence suggests that the field remains conceptually fragmented, with few shared design principles guiding DDT development. This reinforces the motivation behind proposing a structured, hierarchical coordination framework such as DHDT.

The following are the key findings, which built the basis for the morphological box.

The primary distinction in reviewed literature lies in the approach to distribution. One author focused on a singular DDT composed of multiple systems (Ouahabi et al. 2021), while others proposed fully distributed DTs operating in a shared digital environment (Chen et al. 2024; Kim et al. 2025; Vergara et al. 2023; Xia et al. 2024). Also combined approaches were sighted (Baek et al. 2024; Kierans and Pleiter 2024; Zhang et al. 2023). A subset of researchers introduced the concept of an overarching "umbrella DT" governing individual DTs (Abdullahi et al. 2024; Baek et al. 2024). These variations underscore a well-established core concept of DDTs but demonstrate divergence in implementation methodologies.

Governance remains a critical but underexplored area in DDT research. Among the 16 reviewed works, only seven explicitly discussed governance structures, identifying four distinct models:

- **Top-Down:** A centralized DT dictates decisions for all subordinate DTs (Abdullahi et al. 2024; Ouahabi et al. 2021).
- **Centralized-Decentralized:** Governance is delegated to a specialized unit. This means that while the governance functions (e.g., orchestration, synchronization, coordination) are centralized within a dedicated system, this system itself is decoupled from individual DTs and exists as an external, modular layer. The result is a governance approach that is centralized in function but decentralized in placement and architectural integration (Aziz et al. 2023; Vergara et al. 2023).
- **Supervised-Decentralized:** DDTs operate autonomously but under a supervisory mechanism for conflict resolution and intervention (Campo et al. 2024).
- **Decentralized:** DDTs function independently with no centralized oversight (Costantini et al. 2022; Infante et al. 2025).

The absence of governance structures in many sources suggests a need for further research in this domain.

The analysis identified three dominant architectural paradigms:

- **Service-Oriented Architecture (SOA):** Multiple sources described architectures where DTs provide services either through a central entity or in a decentralized manner (Aziz et al. 2023; Baek et al. 2024; Kim et al. 2025; Ouahabi et al. 2021).
- **Distributed Architecture:** Other publications conceptualized DTs as independent, fully functional entities interconnected beyond service exchanges (Campo et al. 2024; Vergara et al. 2024).
- **Monolithic Architecture:** A singular instance where the DT itself was structured monolithically, with only its components exhibiting distributed characteristics (Bonorden et al. 2022).

Some publications describe hierarchies in which higher-level DTs are responsible for coordinating or pushing updates to lower-level DTs. In these cases, the hierarchy primarily serves to schedule or manage the flow of updated control logic, parameters, or trained models. For example, in edge-cloud architectures, global tasks such as model training or policy optimization are executed in the cloud layer and then pushed to the edge DTs for deployment (Ouahabi et al. 2021). Similarly, umbrella-type DTs manage subordinate DTs by regularly distributing updated configurations of ML models, values of predictive maintenance forecasts or control instructions (Abdullahi et al. 2024). These update hierarchies, while effective for deployment and data consistency, do not address behavioral coordination or inter-DT decision-making.

The objectives identified in the reviewed literature were categorized into four primary areas:

- **System Optimization/Control:** Enhancing overall system efficiency, decision-making, and operational coordination (Kim et al. 2025; Vergara et al. 2024; Zhang et al. 2023).
- **Asset Optimization/Control:** Managing individual DTs and their physical counterparts, including predictive maintenance and asset monitoring (Abdullahi et al. 2024; Ouahabi et al. 2021; Xia et al. 2024).
- **Modular Modeling:** Mastering system complexity through digital representations. This approach simplifies the modeling process by decomposing complex systems into smaller, modular digital twins, enabling scalability, reusability, and easier system understanding (Bonorden et al. 2022; Vergara et al. 2024).

- **Shared Learning:** Utilizing distributed machine learning and leveraging collective computational power (Costantini et al. 2022; Infante et al. 2025; Kim et al. 2025; Vergara et al. 2023).

The reviewed literature highlights multiple challenges, categorized as follows:

- **Data Management:** Issues include real-time processing limitations, operational complexity in edge computing, scalability concerns, and ensuring secure data transmission between DTs (Aziz et al. 2023; Chen et al. 2024; Kierans and Pleiter 2024; Xia et al. 2024).
- **Connectivity:** Increased network congestion, latency concerns, and real-time performance degradation due to high data transfer requirements (Abdullahi et al. 2024; Kierans and Pleiter 2024; Vergara et al. 2023).
- **Governance:** Managing heterogeneous systems, ensuring interoperability, and enabling modularity for scalable DDT implementations (Aziz et al. 2023; Baek et al. 2024; Infante et al. 2025).
- **Resource Constraints:** Limited computational power in monolithic systems, high energy consumption, and significant development costs (Abdullahi et al. 2024; Costantini et al. 2022; Ouahabi et al. 2021).

3.2 Hierarchical Digital Twins

The approach to the second literature review was similar to the first. The same databases were queried with the search terms “hierarchical digital twin”, “multi-level digital twin”, “layered digital twin” and “digital twin hierarchy”. The search was conducted explicitly to find literature on behavioral hierarchies, as established in Section 2.3. The searches resulted in 20 potential candidates, which were analyzed in more detail. Only a few of the candidates were concerned with DDTs.

For this secondary review, no emphasis was laid on results already established in the first review. Since only a few of the publications dealt with the distribution of separate DTs, the resulting challenges and tasks were mostly within the space of “Challenges and tasks of DTs”, which are a subgroup of the challenges and tasks identified within the DDT review. Instead, the focus of the deeper analysis was on the hierarchies established by the authors and how these compared to the concept of hierarchical systems, as established in Section 2.3.

The main finding of the review was an apparent lack of behavioral hierarchies in the reviewed literature. Other forms of hierarchies were discovered and are summarized in this chapter.

In Redelinghuys et al. (2020), DTs of individual production cell components were aggregated into higher-level digital twins, forming a layered architecture to manage and structure data exchange. While a master/slave relationship between DTs is mentioned, this is primarily to describe directional data, and potential command flows rather than a detailed behavioral or control dependency. The hierarchy reflects a compositional model, where lower-level DTs represent discrete components (e.g., machines), and higher-level DTs function as aggregates, encapsulating multiple component DTs into structured digital representations of larger subsystems.

Centomo et al. (2021) propose a hierarchical digital twin architecture based on functional abstraction. DTs are organized by system scope ranging from components to entire systems reflecting their roles in monitoring and control. The hierarchy supports top-down coordination and bottom-up data flow but focuses solely on structural and functional relationships. Behavioral dependencies between DTs are not addressed.

Yudin et al. (2023) present a vertically layered hierarchy of DTs reflecting organizational and functional levels within industrial ecosystems. The model defines five levels: product, production, enterprise, industry, and national industrial complex. Each higher-level DT incorporates and extends the capabilities of the lower levels, forming a nested structure. The hierarchy is based on management scope and economic responsibility rather than physical structure. DTs serve distinct roles at each level, from monitoring and optimization to strategic planning and forecasting. The authors emphasize that higher-level DTs cannot exist without the lower layers, reinforcing a compositional and integrative hierarchy.

Beyond the examples discussed above, none of the additional papers identified in our review (Liu et al. 2023; Pan et al. 2024; Pang et al. 2023; Phua et al. 2022; Picano et al. 2024; Ruhe et al. 2023; Zhang et al. 2024) engage with the notion of behavioral hierarchies as defined in Section 2.3. While these works propose various hierarchical structures, ranging from functional control scopes, learning generalization layers, and data aggregation frameworks to compositional system modeling, none describe mechanisms for task delegation, behavioral influence, or dynamic authority assignment between DTs. Hierarchy is consistently treated as structural, semantic, or organizational, not as a means to establish inter-DT behavioral relationships. This consistent absence across otherwise diverse DT architectures highlights a significant conceptual and practical gap in current hierarchical digital twin research.

3.3 Review Synthesis

The first review revealed a broad range of challenges encountered in DT development. Many of these are tackled by the use of DDTs. Issues such as limited system capacity or high network load were addressed through edge computing and distributed deployment strategies. However, these solutions introduced new challenges, particularly in the area of governance. Interoperability, scalability, modularity, and the management of heterogeneous systems were repeatedly identified as critical issues.

One emerging governance challenge is orchestration. To reduce network congestion, many authors proposed minimizing communication between DTs. While effective in isolation, this approach increases the autonomy of individual DTs and risks undermining global optimization goals. Without some form of central coordination, DTs may pursue conflicting local objectives, possibly degrading overall system performance.

Hierarchical structures offer a potential solution. In a hierarchical DDT system, a supervisory DT could be assigned authority over others, enabling coordinated behavior across the network. This allows for the alignment of local DT actions with a global objective, preventing conflicts and supporting more coherent system-wide optimization.

The second review found no evidence of behavioral hierarchies that would be necessary to implement such coordination. While structural and functional hierarchies were observed, often focused on data aggregation or system decomposition, none of the reviewed works described mechanisms for behavioral control, task delegation, or dynamic prioritization between DTs.

Together, the two reviews reveal a significant gap in current research: the absence of inter-DT orchestration mechanisms capable of resolving conflicts between local and global objectives. DTs are generally treated as isolated components that communicate via data exchange but operate independently. In some cases, the potential for goal conflict is acknowledged, but no solution is proposed.

To achieve coherent behavior in DDT systems, and especially in DHDTs, a mechanism for inter-DT control is essential. Only through such structures can the full benefits of distributed architectures be realized without compromising on system-wide optimization goals.

4 DISTRIBUTED HIERARCHICAL DIGITAL TWIN

In response to the gaps identified in previous sections, this chapter outlines a set of requirements and conceptual elements for a potential DHDT system. Such a system should enable dynamic orchestration and behavioral coordination across modular DTs. Figure 1 illustrates a conceptual architecture with generic examples of possible DT components.

A DHDT system should be composed of modular DTs, each representing a distinct asset, process, or subsystem. These DTs would need to conform to a standardized communication and behavioral architecture to support plug-and-play compatibility. The system should also be dynamically scalable, allowing DTs to be added or removed during runtime without requiring global reconfiguration. This would depend on a common interface definition and a shared interaction protocol that facilitate recognition, integration, and cooperation among DTs.

Each DT is expected to encapsulate its own local optimization goal (e.g., energy minimization, throughput maximization), which governs its autonomous behavior. A global goal might be introduced into the system by a designated DT, either the one with the highest granted dominance score or one manually selected by the system designer. In this context, dominance refers to the ability of one DT to influence or override aspects of another DT's behavior or objectives. Rather than acting solely as a coordination anchor, the global goal could actively shape the behavior of all DTs that permit access. If access is restricted, DTs would default to their local goals; otherwise, the global goal could override and reorient individual behaviors. Changes to the global goal, or targeted influence commands from dominant DTs, could prompt subordinate DTs to re-evaluate their goals or configurations.

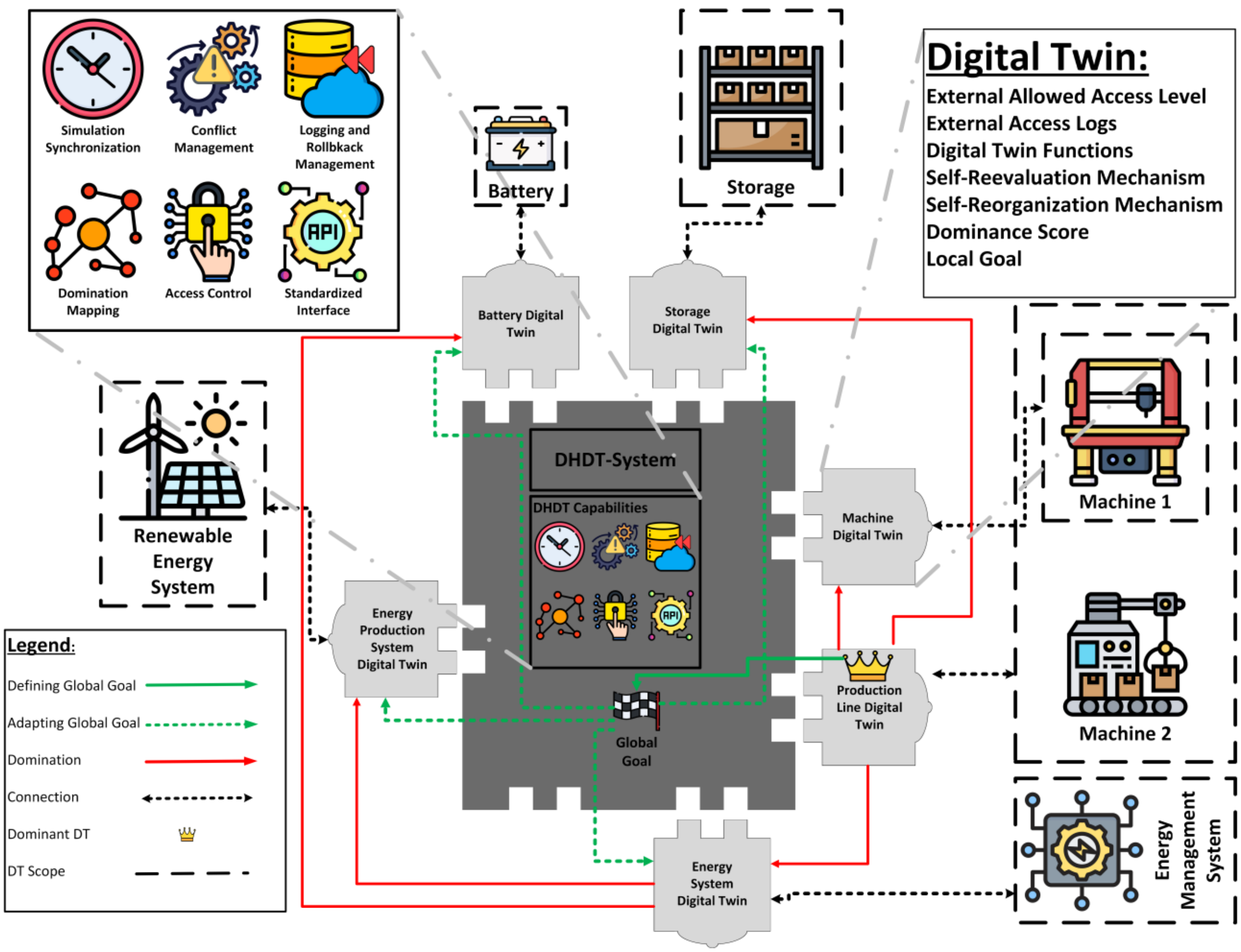

Figure 1: Distributed, hierarchical digital twin.

Hierarchies within a DHDT system do not need to be static but instead could emerge dynamically based on runtime conditions. A DT may attempt to assert dominance over another based on contextual relevance, granted access, or optimization priority. Dominance could allow the influencing DT to adjust goals, trigger functions, or reconfigure behaviors in the subordinate. Only one DT should dominate another at a time. In case of conflict, dynamic resolution mechanisms, such as weighted importance scoring, predicted impact of outcomes, or negotiation protocols, might be required to determine the dominant party. Importantly, dominance should be transitional: if unintended or harmful behavior is detected, the affected DT could revoke access, triggering a re-evaluation of hierarchy and a potential rollback to a previous stable state. Access between DTs should be regulated by clearly defined levels, ranging from read-only data sharing to full behavioral control or assimilation. These access levels could be set during development or adjusted dynamically during operation. For instance, if a DT identifies negative effects resulting from another DT's command, it might automatically reduce access privileges or initiate a trust reevaluation. With appropriate access, DTs could read internal states, invoke simulations, or execute embedded optimization routines of other DTs to inform their own decisions or improve overall system performance.

To ensure transparency and traceability, a shared domination interface should maintain a real-time registry indicating which DTs are currently dominated, by whom, and at what level of access. This registry would help prevent conflicting control attempts, support debugging, and provide a foundation for human oversight in complex, multi-domain environments. The visibility of current influence relationships could help system designers and automated governance layers identify conflict patterns or improve dominance assignments.

A simulation coordination system should also be part of the DHDT architecture. Its purpose would be to ensure proper synchronization between the simulation models embedded within different DTs. Such a system should be capable of linking various simulation types, including both discrete-event and continuous models, and support the collective initialization of co-simulation runs involving multiple or all DTs. In scenarios where optimistic simulation approaches are applied, the coordination system would also need to handle rollback mechanisms to maintain consistency across distributed simulations.

All access actions, including goal modifications, data retrieval, command execution, and simulation initiation, should be logged with time stamps, actor identity, and relevant outcome metadata. This logging is essential not only for accountability and security auditing, but also for enabling system rollbacks if unintended behaviors arise. In addition, these logs could serve as valuable historical inputs for simulation and training, allowing for reanalysis of past decision sequences under varying global and local goals.

5 CONCLUSION

This paper set out to review the current state of research on Digital Twins, with a specific focus on their distributed and hierarchical forms. Through two targeted literature reviews, we identified a notable gap: while structural and functional hierarchies are frequently addressed, the concept of behavioral hierarchies, where DTs coordinate decisions, delegate tasks, or resolve conflicting goals, remains largely unexplored. This lack of mechanisms for behavioral orchestration presents a major limitation for scaling DTs into more complex, interconnected systems.

In light of this gap, we introduced the concept of Distributed, Hierarchical Digital Twins and outlined a preliminary architecture that illustrates requirements for such behavioral coordination to be realized. These early ideas include dynamic goal negotiation, access-based control, and context-aware hierarchy formation between DTs. However, this architecture should be understood as a starting point for discussion, not a fully developed solution.

Further research is needed to explore how such systems could be implemented, governed, and validated in practice. This also concerns questions of dominance and orchestration. Key challenges include defining lightweight coordination protocols, ensuring scalability, and developing methods for conflict resolution between competing optimization objectives. The findings of this paper are intended to serve as a foundation for these next steps, and to encourage deeper investigation into the behavioral dimension of DT interoperability.

REFERENCES

Abdullahi, I., S. Longo, and M. Samie. 2024. "Towards a Distributed Digital Twin Framework for Predictive Maintenance in Industrial Internet of Things (IIoT)". *Sensors* 24(8):2663.

Aivaliotis, P., K. Georgoulias, and G. Chryssolouris. 2019. "The Use of Digital Twin for Predictive Maintenance in Manufacturing". *International Journal of Computer Integrated Manufacturing* 32(11):1067–1080.
Aziz, A., S. S. Chouhan, O. Schelén, and U. Bodin. 2023. "Distributed Digital Twins as Proxies-Unlocking Composability and Flexibility for Purpose-Oriented Digital Twins". *IEEE Access* 11:137577–137593.
Baek, M.-S., E. Jung, Y. S. Park, and Y.-T. Lee. 2024. "Federated Digital Twin Implementation Methodology to Build a Large-Scale Digital Twin System". In *2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, June 19th-21st, Toronto, ON, Canada, 1-2.
Barykin, S. Y., A. A. Bochkarev, O. V. Kalinina, and V. K. Yadykin. 2020. "Concept for a Supply Chain Digital Twin". *International Journal of Mathematical, Engineering and Management Sciences* 5(6):1498–1515.
Bonorden, L., M. Frerichs, M. Riebisch, S. von Riegen, F. Hartke, R. Herzog, *et al*. 2022. "Decision-Making About Federated Digital Twins – How to Distribute Information Storage and Computing". In *Modellierung 2022*, June 27th-July 1st, Hamburg, Germany, 49-64.
Campo, G., M. Russo, C. Santoro, and F. F. Santoro. 2024. "Enhancing Robotic Systems Through a Distributed Multi-Agent Digital Twin Environment". In *2024 IEEE Conference on Pervasive and Intelligent Computing (PICom)*, November 5th-8th. Boracay Island, Philippines, 75–82.
Centomo, S., A. Avogaro, M. Panato, C. Tadiello, and F. Fummi. 2021. "A Design Methodology of Multi-Level Digital Twins". In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, March 10th-12th, Valencia, Spain, 961–966.
Chen, Z., W. Yi, A. Nallanathan, and J. A. Chambers. 2024. "Distributed Digital Twin Migration in Multi-Tier Computing Systems". *IEEE Journal of Selected Topics in Signal Processing* 18(1):109–123.
Cheng, G., W. Wei, and Z. Liu. 2021. "Research on Intelligent Operation and Maintenance System of Battery Based on Digital Twin". In *2021 2nd International Conference on Computer Engineering and Intelligent Control (ICCEIC),* November 12th-14th, Chongqing, China, 2021, 154-157.
Costantini, A., G. Di Modica, J. C. Ahouangonou, D. C. Duma, B. Martelli, M. Galletti, *et al*. 2022. "IoTwins: Toward Implementation of Distributed Digital Twins in Industry 4.0 Settings". *Computers* 11(5):67.
Dietz, M., and G. Pernul. 2020. "Digital Twin: Empowering Enterprises Towards a System-of-Systems Approach". *Business & Information Systems Engineering* 62(2):179–184.
Fatemeh, T., H. M. Adel, and J. Nathalie. 2023. "CODIT 2023 Modeling of Smart Batteries for the Realization of a Digital Twin Prototype". In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT),* July 3rd-6th, Rome, Italy, 2396-2401.
Gao, B., H. Liu, H. Su, Z. Hong, C. Ji, and M. Zhao. 2022. "Design and Typical Application of Digital Twin Architecture in Smart Grid". In *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE),* April 15th-17th, Hangzhou, China, 2140–2144.
Hammar, K., and R. Stadler. 2023. "Digital Twins for Security Automation". In *Proceedings of IEEE/IFIP Network Operations and Management Symposium*, May 8th-12th, Miami, FL, USA, 1–6.
Infante, S., J. Robles, C. Martín, B. Rubio, and M. Díaz. 2025. "Distributed Digital Twins on the Open-Source OpenTwins Framework". *Advanced Engineering Informatics* 64:102970.
Kierans, D., and D. Pleiter. 2024. "Realising Distributed Digital Twins within Federated Digital Infrastructures". In *1st International Workshop on Distributed Digital Twins*, June 17th, Groningen, Netherlands, 1-12.
Kim, Y.-J., H. Kim, B. Ha, and W.-T. Kim. 2025. "Federated Digital Twins: A Scheduling Approach Based on Temporal Graph Neural Network and Deep Reinforcement Learning". *IEEE Access* 13:20763–20777.
Kuehner, K. J., R. Scheer, and S. Strassburger. 2021. "Digital Twin: Finding Common Ground – A Meta-Review". *Procedia CIRP* 104:1227–1232.
Lehner, C., A. Padovano, C. Zehetner, and G. Hackenberg. 2024. "Digital Twin and Digital Thread Within the Product Lifecycle Management". *Procedia Computer Science* 232:2875–2886.
Li, L., F. Gu, H. Li, J. Guo, and X. Gu. 2021. "Digital Twin Bionics: A Biological Evolution-Based Digital Twin Approach for Rapid Product Development". *IEEE Access* 9:121507–121521.
Liu, X., C. Qiu, J. Shi, J. Huang, C. Zhu, Z. Ni, M. Zhu, and T. Liu. 2023. "A Digital Twin Modeling Method for Production Resources of Shop Floor". *The International Journal of Advanced Manufacturing Technology* 128(1-2):743–761.
Mihai, S., M. Yaqoob, D. V. Hung, W. Davis, P. Towakel, M. Raza, *et al*. 2022. "Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects". *IEEE Communications Surveys & Tutorials* 24(4):2255–2291.
Ouahabi, N., A. Chebak, M. Zegrari, O. Kamach, and M. Berquedich. 2021. "A Distributed Digital Twin Architecture for Shop Floor Monitoring Based on Edge-Cloud Collaboration". In *2021 Third International Conference on Transportation and Smart Technologies,* May 27th-28th, Tangier, Morocco, 72–78.
Pan, Y., R. Y. Zhong, T. Qu, L. Ding, and J. Zhang. 2024. "Multi-Level Digital Twin-Driven Kitting-Synchronized Optimization for Production Logistics System". *International Journal of Production Economics* 271:109176.
Pang, J., P. Zheng, S. Li, and S. Liu. 2023. "A Verification-Oriented and Part-Focused Assembly Monitoring System Based on Multi-Layered Digital Twin". *Journal of Manufacturing Systems* 68:477–492.
Phua, A., C. Davies, and G. W. Delaney. 2022. "A Digital Twin Hierarchy for Metal Additive Manufacturing". *Computers in Industry* 140:103667.

Picano, B., M. Becattini, L. Carnevali, and E. Vicario. 2024. "Democratized Learning Enabling Multi-Level Digital Twin Model Integration". In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA),* September 10th-13th, Padova, Italy, 1–8.

Poechgraber, G., S. Bougain, B. Wallner, G. Bohaty, T. Trautner, and F. Bleicher. 2023. "Introduction of a Digital Twin for the Product Development Phase". In *2023 International Conference on Engineering Management of Communication and Technology (EMCTECH),* October 16th-18th, Vienna, Austria, 1–6.

Rayhana, R., L. Bai, G. Xiao, M. Liao, and Z. Liu. 2024. "Digital Twin Models: Functions, Challenges, and Industry Applications". *IEEE Journal of Radio Frequency Identification* 8:282–321.

Redelinghuys, A. J. H., K. Kruger, and A. Basson. 2020. "A Six-Layer Architecture for Digital Twins with Aggregation". In *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future: Proceedings of SOHOMA 2019,* edited by T. Borangiu, 171–182. Cham: Springer International Publishing AG.

Ruhe, S., K. Schaefer, S. Branz, S. Nicolai, P. Bretschneider, and D. Westermann. 2023. "Design and Implementation of a Hierarchical Digital Twin for Power Systems Using Real-Time Simulation". *Electronics* 12(12):2747.

Shao, G., S. Frechette, and V. Srinivasan. 2023. "An Analysis of the New ISO 23247 Series of Standards on Digital Twin Framework for Manufacturing". In *Proceedings of the ASME 2023 18th International Manufacturing Science and Engineering Conference. Volume 2: Manufacturing Equipment and Automation; Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability,* June 12th-16th, New Brunswick, New Jersey, USA, 1-10.

Strelets, D. Y., S. A. Serebryansky, and M. V. Shkurin. 2020. "Concept of Creation of a Digital Twin in the Uniform Information Environment of Product Life Cycle". In *Proceedings of 2020 Thirteenth International Conference Management of Large-Scale System Development (MLSD),* September 28th-30th, Moscow, Russia, 1–4.

van Steen, M., and A. S. Tanenbaum. 2016. "A Brief Introduction to Distributed Systems". *Computing* 98(10):967–1009.

Vergara, C., R. Bahsoon, G. Theodoropoulos, W. Yanez, and N. Tziritas. 2023. "Federated Digital Twin". In *2023 IEEE/ACM 27th International Symposium on Distributed Simulation and Real Time Applications (DS-RT),* October 4th-5th, Singapore, 115–116.

Vergara, C. R., G. Theodoropoulos, R. Bahsoon, W. Yanez, and N. Tziritas. 2024. "Federated Digital Twins as an Enabling Technology for Collaborative Decision-Making". In *Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation,* June 24th-26th, Atlanta, Georgia, USA, 67–68.

Veríssimo, P. 2001. *Distributed Systems for System Architects*. Boston, MA: Springer.

Xia, Y., X. Liu, Y. Zhao, and Y. Wang. 2024. "FCLA-DT: Federated Continual Learning with Authentication for Distributed Digital Twin-Based Industrial IoT". *Journal of Communications and Information Networks* 9(4):362–373.

Yudin, D., T. Yudina, and E. Yudina. 2023. "Agent-Based Modeling in Multi-Level Industrial Ecosystems Development". *Revista Relações Internacionais do Mundo Atual* 4(42):703–716.

Zhang, H., Q. Yan, Y. Qin, S. Chen, and G. Zhang. 2023. "A Novel Approach of Resource Allocation for Distributed Digital Twin Shop-Floor". *Information* 14(8):458.

Zhang, L. 2023. "Application of Digital Twins in Improving Efficiency and Reducing Cost in Maintenance of Industrial Building Complexes in Universities". In *2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE)*, October 27th-29th, Yunlin, Taiwan, 343–346.

Zhang, X., Y. Yang, X. Zhang, Y. Hu, H. Wu, M. Li, *et al*. 2024. "A Multi-Level Digital Twin Construction Method of Assembly Line Based on Hybrid Worker Digital Twin Models". *Advanced Engineering Informatics* 62:102597.

AUTHOR BIOGRAPHIES

AATU KUNNARI is a research associate at the Department of Information Technology in Production and Logistics at the Ilmenau University of Technology. He holds a M.Sc. and B.Sc. in business information systems from the Ilmenau University of Technology. His research interests include industrial simulation, distributed simulation and digital twins. His email address is aatu-juhani.kunnari@tu-ilmenau.de.

STEFFEN STRASSBURGER is a professor at the Ilmenau University of Technology and head of the Group for Information Technology in Production and Logistics. Previously he was head of the "Virtual Development" department at the Fraunhofer Institute in Magdeburg, Germany and a researcher at the DaimlerChrysler Research Center in Ulm, Germany. He has been involved with simulation for more than 25 years. He holds a Doctoral and a Diploma degree in Computer Science from the University of Magdeburg, Germany. His research interests include distributed simulation, automatic simulation model generation, and general interoperability topics within the digital factory and Industry 4.0 context. His email address is steffen.strassburger@tu-ilmenau.de. The website of his department is https://www.tu-ilmenau.de/itpl/.