Digital Twins Ecosystems for Intelligent Applications in Healthcare

Samuele Burattini
PhD in Computer Science and Engineering
University of Bologna

Abstract

The digitalization of healthcare is essential to improve efficiency, traceability, and decision-making. Digital Twins offer virtual representations of physical assets, yet healthcare involves many heterogeneous and interconnected elements, posing challenges for integrated systems. This thesis, a collaboration with AUSL Romagna, proposes architectural and engineering solutions that promote modularity, interoperability, and semantic descriptions of Digital Twin to support their integration and management within *Digital Twin Ecosystem*. Inspired by the Web, it introduces a *Hypermedia Web of Digital Twins* prototype enabling the development of heterogeneous Digital Twin Ecosystems reusing Web standards and principles. Finally, the thesis investigates the use of Multi-Agent Systems to implement trustworthy intelligent applications in Digital Twin Ecosystems, clarifying their relationship with Digital Twins. The approach is validated through healthcare application scenarios.

1 Introduction

In the last decade, we have assisted to a significant transformation driven by the integration of advanced digital technologies such as the Internet of Things (IoT), Artificial Intelligence (AI) and Machine Learning (ML) to improve the efficiency and automation of processes across manufacturing and service industries. This digital transformation, often referred to as the *Fourth Industrial Revolution* or *Industry 4.0*, is now becoming a pervasive trend impacting even less technology-driven sectors such as agriculture and healthcare [1]. In the healthcare sector, the integration of digital technologies is reshaping the way care is delivered, with a focus on improving patient outcomes, enhancing operational efficiency, and enabling personalized medicine [2].

In this context, the concept of Digital Twin (DT) [3] has emerged as a powerful abstraction to model and represent physical assets by creating a software replica which is continuously updated with data from the physical world. Complex domains such as healthcare, however, are characterized by a variety of heterogeneous assets whose interrelations are crucial to provide a complete and accurate representation of the physical context [4]. This thesis explores the engineering challenges in the development of *Digital Twin Ecosystems (DTEs)* integrating heterogeneous DTs connected by dynamic relationships. The work further propose the DTE as a layer of abstraction to support the development of intelligent applications which can exploit its capabilities to acquire a comprehensive view of the physical world and automate processes [5]. To address these challenges, this work proposes architectural and engineering solutions for the development of DTs focusing on modularity and interoperability. This approach is complemented by the introduction of (semantic) descriptions for DTs that can support both the operational management and integration challenges in DTEs. Through an approach inspired by the hypermedia nature of the Web [6] and the Web of Things [7] a prototype for a *Hypermedia Web of Digital Twins (HWoDT)* which can be used to implement heterogeneous DTEs is developed [8]. Finally, the thesis investigates the use of Multi-Agent Systems (MAS) [9] as a way to implement trustworthy intelligent applications in DTE, defining their relationship with DTs and contributing to the engineering of MAS based on the Belief-Desire-Intention model [10]. The proposed approach is validated through the analysis and development of application scenarios in the healthcare domain.

2 Digital Twin Engineering

Recalling the definition of DT from [11]:

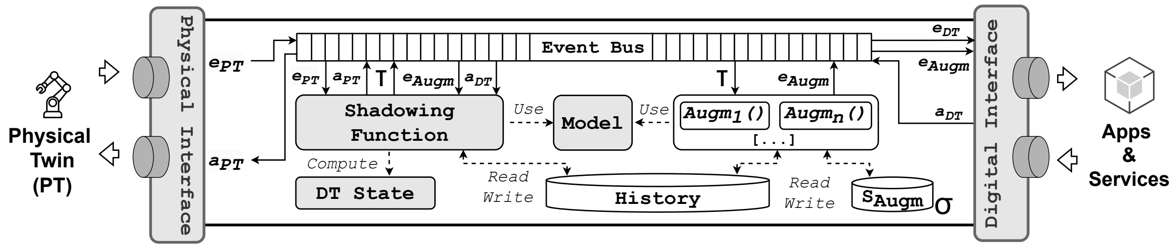


Figure 1: Event-Driven architecture of a DT to support and enable effective Augmentation function management and execution.

> *A Digital Twin (DT) is a comprehensive software representation of an individual Physical Asset. It includes the properties, conditions, and behavior(s) of the real-life asset through models and data. A DT is a set of realistic models that can simulate an asset's behavior in the deployed environment. The DT represents and reflects its physical twin and remains its virtual counterpart across the asset's entire lifecycle.*

This thesis contributes to the engineering of such DTs, focusing on the ability to: *(i)* integrate with heterogeneous data sources and services; *(ii)* manage the synchronization lifecycle effectively; *(iii)* encapsulate augmented features that enhance the capabilities of the mirrored Physical Assets (PAs)

Starting from the principles outlined by the 5D model [12], and incorporating insights from both recent academic research [5, 13] this thesis proposes an abstract DT architecture composed of (Figure 1):

- *Physical Interface (PI):* tasked with both the digitalization process and the ongoing synchronization of the DT and PA throughout their lifecycle based on its characteristic cyber-physical nature and the supported protocols and data formats (e.g., HTTP and JSON, MQTT and binary);
- *Digital Interface (DI):* complementing the PI, it manages the routing of DT's internal variations and events directed towards external digital entities and consumers ensuring the DT's interaction, interoperability and observability;
- *Model (M):* Defines the DT's behavior through the digitalization process together with augmented functionalities. The *shadowing* process is responsible for handling events from both the PI and DI to model and replicate the state of the asset, while *augmentation* [11] functions extends functionalities of the PA through additional features and capabilities supported by the software nature of the twin (e.g., simulation, machine learning inference models).

General-purpose DT cloud platforms serve primarily as data repositories, integrating with cloud storage and IoT services, where data acquisition and pre-processing are managed externally, and platform-level protocols constrain the shadowing process and client interactions. Industrial DT platforms from companies such as Siemens, GE, Bosch, and IBM are typically domain-specific, tightly integrated with their own assets, and favor proprietary solutions, resulting in limited interoperability due to a lack of common modeling standards. In contrast, the proposed event-driven approach enables both the PI and DI to capture and forward events from physical and digital domains to the M, allowing DTs to function as independent software entities with configurable interfaces for flexible integration and interaction.

The architectural proposal has been implemented in an open-source software framework: the White Label Digital Twins (WLDT) [14] library[1].

2.1 Modular Interoperability in Digital Twin Interfaces

DTs must connect to diverse PAs using a variety of protocols and legacy infrastructures, while also serving multiple digital consumers that may require different data representations or interaction paradigms[15]. Achieving this requires both support for existing standards and architectural flexibility when uniformity cannot be imposed.

We address this problem through a modular interface design that decouples the DT core from communication concerns. On the physical side, *Physical Adapters (PhAs)* interface with the PA, producing *Physical*

[1] `https://wldt.github.io/`

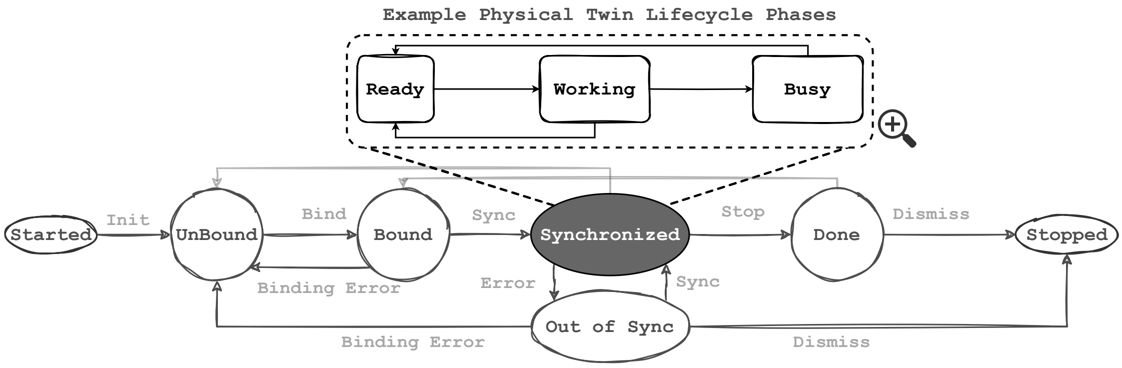


Figure 2: Schematic representation of the DT lifecycle.

Asset Descriptions (PADs) that describe properties, actions, and events independently of protocol-specific constraints. Thus, protocol or payload changes do not affect the internal DT model.

On the digital side, *Digital Adapters (DiAs)* expose the DT state and services to different classes of applications. This enables adaptation to legacy systems or domain-specific platforms without modifying the physical integration. To assist discovery and automation in distributed DTEs, the capabilities offered through DiAs can be described using a *Digital Twin Description (DTD)*, intended for consumption by external systems.

2.2 Managing the Digital Twin Lifecycle

The lifecycle of a DT is crucial to ensure continuous and trustworthy alignment with its PA, particularly regarding reflection and entanglement [11, 5]. Although recent studies recognize this importance [16], existing lifecycle proposals are mostly domain-specific and lack concrete guidance on how internal DT components maintain a consistent state [17].

To address this, we adopt a structured lifecycle that explicitly models synchronization between the DT and the PA (Figure 2). After initialization (`Started`), the DT prepares for communication in `Unbound`. Once the necessary information from the PA is confirmed, it transitions to `Bound`. The `Synchronized` phase is reached only when communication quality and completeness allow accurate computation of the DT state. If these conditions degrade, the DT enters `OutOfSync`. When the DT is no longer needed, it moves to `Done` and can finally be `Stopped`. Corrective actions may return it to `Unbound` at any time.

A major challenge emerges from the variability in PA operational behavior. Communication delays or temporary data unavailability may be intentional (e.g., `Idle`, `Rebooting`, `Overloaded`) rather than failures. Therefore, synchronization expectations must adjust dynamically to avoid false anomaly detection and preserve cyber-physical consistency.

This structured lifecycle improves:

- *Cyber-physical awareness* through explicit tracking of synchronization status;
- *Decision-making* by ensuring state updates accurately reflect physical conditions;
- *Adaptability* across heterogeneous domains and operational dynamics.

Open challenges remain, particularly in reliably identifying the PA operational context and defining tolerances for synchronization failure. However, this lifecycle model provides a robust foundation for engineering reliable and maintainable DT software in dynamic environments.

2.3 Modeling Augmentation Functionalities

PAs are constrained by their physical nature, while DTs can be enhanced and updated through software to introduce new capabilities over time This property, known as *augmentation* [11], enables DTs to support advanced features such as interoperability improvements, security enhancements, higher-level actions, and intelligent behaviors powered by data-driven models and AI. However, augmentation is often described conceptually, with implementations largely domain-specific and without a clear reference model, which may compromise *fidelity* to the PA [13].

The proposal is to model such features that the DT can expose through service interfaces relying on internal models to analyze the DT state and history (e.g., anomaly detection or predictive capabilities) as *augmentation functions*. Such functions can differ along two main dimensions. First, its **triggering mechanism** defines whether the function responds to changes in the DT state, the output of other augmentation processes, or scheduled temporal conditions. Second, its **state management** determines whether the function is stateless or maintains an internal state that evolves in parallel with the DT lifecycle, enabling continuous or long-running processing.

Augmentation functions hence operate in an event-driven manner and may generate new internal events that feed back into the core synchronization logic. When relevant, their results also become externally accessible through the DI. This approach avoids direct coupling with the physical world: all augmentation is mediated by the DT internal model, ensuring alignment with the cyber-physical representation.

Depending on deployment needs and performance constraints, augmentation may be:

- **Internal**: functions executed within the DT process, ensuring low latency and tight integration;
- **External**: functions executed by independent services, supporting distribution and access to specialized resources;
- **Hybrid**: a combination of both, offering flexible scaling across heterogeneous workloads.

Internal augmentation is suitable for real-time processing tied to current physical conditions. External augmentation is preferred when leveraging advanced analytics or computational offloading. Hybrid architectures blend these strengths—combining responsiveness with scalable and resource-intensive functionalities. This unified architectural view makes augmentation a first-class element of DT engineering, clarifying its role in enhancing capabilities while preserving fidelity and ensuring transparent integration with the core synchronization mechanisms of the DT.

3 Digital Twin Ecosystems

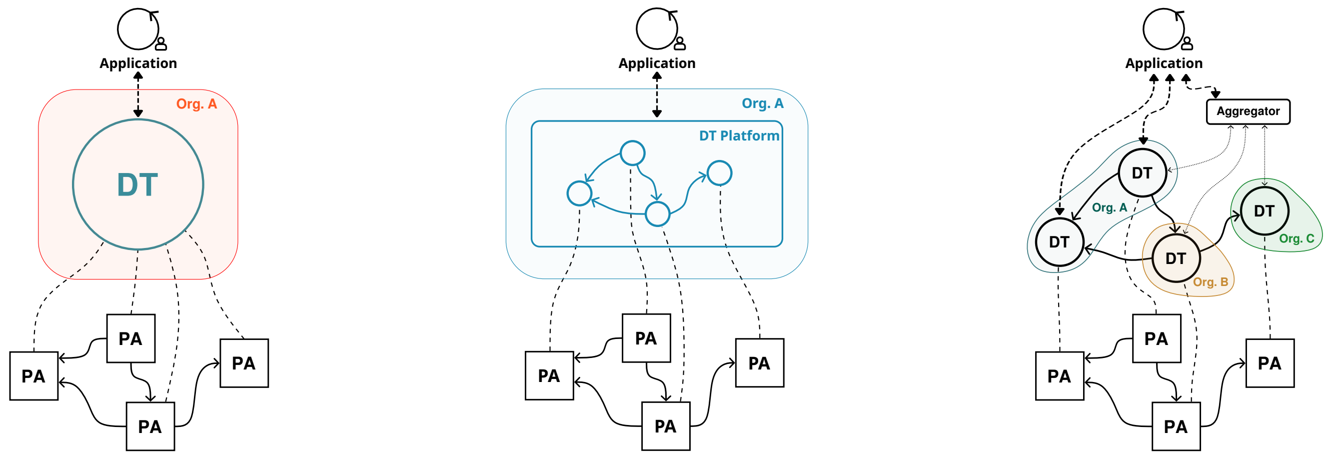
DTEs emerge as a natural evolution of the DT concept, driven by the need of breaking out of the silos of individual DTs to better represent the interactions that occur in the physical world among different PAs which can rarely be considered completely in isolation. This is particularly relevant when taking the perspective of using DTs as system-wide abstractions to design Cyber-Physical System (CPS) [18]. For the scope of this thesis, the following definition of Digital Twin Ecosystem is adopted:

> *A Digital Twin Ecosystem is a dynamic set of Digital Twins, each modeling a Physical Asset, whose meaningful relationships within a target context make it valuable to consider their collective evolution to accurately represent the mirrored portion of the physical world.*

This definition highlights three key aspects of Digital Twin Ecosystems:

- **Dynamic set of Digital Twins**: DTEs are not static constructs; they can evolve over time as new DTs are added or removed, reflecting changes in the physical environment or system requirements.
- **Meaningful relationships**: The interactions and dependencies among the constituent DTs are crucial. These relationships can be functional, spatial, temporal, or based on data exchange, and they define how the DTs influence each other within the ecosystem.
- **Collective evolution**: The value of a DTE lies in its ability to represent the collective behavior of its constituent DTs. This holistic view enables new forms of interactions that support processes that consider the interdependencies among multiple PAs.

The characteristics and functional requirements identified for DTEs lead to different trade-offs concerning *(i)* the ability to evolve to accommodate changes in the domain of interest, *(ii)* loose coupling of the member DTs, and *(iii)* implementation of the ecosystem functionalities (query, discovery, observation of DTs in the ecosystem). This can lead to different architectural approaches emerging from the state of the art of DTE research and technologies (Figure 3).



(a) **Monolithic** Ecosystem

(b) **Homogeneous** DTE

(c) **Heterogeneous** DTE

Figure 3: Three architectural approaches for DTEs: in 3a a single DT models the whole scenario; in 3b a single platform is used to build and deploy all the DTs in the ecosystem; in 3c maximum flexibility is allowed, possibly reusing existing DTs under a common interface.

Monolithic Ecosystem A straightforward approach is to model the entire context as a single, monolithic DT. Even complex entities (e.g., a city) can be represented as one DT, with internal components modeled explicitly [19]. The main advantage is architectural simplicity: all data flows into a single system built with a consistent stack (Figure 3a), enabling unified simulations and management. However, updating or extending the model can be challenging, as even minor changes may require modifying the whole DT. This approach is less suitable for dynamic, heterogeneous, or multi-stakeholder scenarios and limits technological diversity.

Homogeneous DTE A *Homogeneous* DTE uses a general-purpose platform supporting multiple DTs (Figure 3b). All DTs share the same modeling and implementation technology, simplifying ecosystem services and updates. This approach highlights the individuality of entities while maintaining ease of management. However, reliance on a single platform can introduce vendor lock-in and modeling constraints. An example is Azure Digital Twins, which uses a *Twin Graph*[2] to represent relationships among DTs modeled with a uniform Digital Twin Definition Language (DTDL).

Heterogeneous DTE This approach introduces a common interoperability layer over technologically diverse DTs, enabling reuse and uniform access (Figure 3c). DTs are standalone components with specific models and technologies, accessed directly or via aggregators. It suits dynamic, open ecosystems with loosely coupled components and multiple stakeholders. DTs can belong to multiple ecosystems, increasing flexibility. However, each DT must be mapped to a shared metamodel and interface, which may reduce expressivity. Direct access to original DTs remains possible for specialized needs. Implementing ecosystem services is more complex than in previous approaches and may require distributed techniques (e.g., polystores, federated or link-traversal queries) or middleware to aggregate and manage DTs.

3.1 Proposing the Hypermedia Web of Digital Twins

The proposed approach aims to enable seamless integration of existing DTs, regardless of their underlying technologies, by hiding heterogeneity behind a *uniform interface* based on Web protocols and standards. An explicit semantic layer allows DTs to uniformly describe their state, features, and services. The term *Hypermedia Web of Digital Twins (HWoDT)* highlights the hypermedia-driven nature of the approach, following the Hypermedia as the Engine of Application State (HATEOAS) principle of REpresentational State Transfer (REST) [6].

This interoperability layer enables navigation and interaction with heterogeneous DTs. To provide ecosystem-level functionalities, a *WoDT Platform* is introduced as both a scope boundary for ecosystem

[2] `https://learn.microsoft.com/en-us/azure/digital-twins/concepts-twins-graph`

membership and an aggregation layer for querying, observing, and exploiting services across distributed DTs.

As illustrated in Figure 3c, the resulting architecture is composed of DTs: *(i)* created using heterogeneous technologies, *(ii)* implementing the uniform interface through *adapters*, *(iii)* connected by relationships that reflect the physical ones, *(iv)* and that are aggregated into DTEs by being registered to one or multiple WoDT Platforms.

To integrate heterogeneous DTs, each DT must expose a uniform interface aligned with HWoDT requirements. Achieving standard semantic representations remains an open challenge. This work adopts a pragmatic approach, showing that uniform semantic representations enable interoperability: each DT should provide both *data* and *affordances* (i.e., possible actions) via hypermedia representations, following the HATEOAS principle. This leads to distinguish two logically different representations:

- *Digital Twin Description (DTD)*: Inspired by the Web of Things (WoT) Thing Description, the DTD holds metadata and affordances, enabling consumers to understand the DT model, as well as which services the DT exposes and how to access them. It provides management metadata, including persistent, globally unique Uniform Resource Identifier (URI) identifiers for both the DT and its associated PA, links to the ecosystem (DTE) it belongs to, and describes the available properties, relationships, events, and actions. The DTD also specifies the API and protocol bindings for interaction, using hypermedia controls in line with REST principles. While it may evolve with updates, it remains mostly static as it describes identity and interface, not real-time state.
- *Digital Twin Knowledge Graph (DTKG)*: The Digital Twin Knowledge Graph (DTKG) represents the live state of the DT using a Knowledge Graph (KG) based on Resource Description Framework (RDF) triples. It semantically encodes domain knowledge, supporting state observation, querying, and discovery of relationships between DTs. The DTKG includes current property values, relationships with other DTs, and context-dependent available actions, but excludes transient events. By leveraging domain-specific ontologies and Linked Data principles, the DTKG enables distributed navigation and a common interpretation of DT data across the ecosystem.

A prototype for the HWoDT has been implemented and is available as open source[3], supporting the integration of three different DT platforms through configurable, reusable *adapters*, to facilitate onboarding. Preliminary performance evaluations show promising results and confirm that the decoupling from different platforms can reduce complexity in the interaction with multiple DTs [20].

The HWoDT approach distinguishes itself from existing DT interoperability frameworks by leveraging REST principles and Semantic Web technologies to enable integration of heterogeneous DTs within open ecosystems. Unlike industrial standards (such as OPC-UA, AAS, AML, and W3C WoT) wich often target specific domains, enforce homogeneous technology stacks, or limit expressiveness in querying and relationship modeling, the HWoDT provides a uniform, hypermedia-driven interface and semantic layer that supports advanced queries, reasoning, and cross-domain knowledge representation. This approach complements rather than competes with existing standards, allowing integration atop established DT implementations and bridging gaps in interoperability, especially outside industrial contexts. Compared to closed DT platforms like Azure Digital Twins or open-source solutions such as Eclipse Ditto, HWoDT avoids vendor lock-in, supports dynamic relationships, and enables direct, up-to-date access to DT state via the DTKG. While sharing some motivations with other web-based DT initiatives and WoT mashups, HWoDT uniquely combines semantic descriptions, live state representation, and open standards to facilitate scalable, flexible, and expressive DT ecosystems across diverse domains.

4 Intelligent Applications with Multi-Agent Systems

AI is increasingly driving Healthcare 4.0, supporting both clinical and administrative processes. Early applications focused on rule-based decision support, while advances in ML, Deep Learning (DL), and the availability of large datasets have enabled data-driven approaches for image-based diagnosis, genome interpretation, biomarker discovery, risk prediction, and IoT-based patient monitoring [21]. Emerging technologies such as Large Language Models (LLMs) and Generative Artificial Intelligence (GenAI) further expand applications, including clinical documentation automation, patient interaction, data augmentation,

[3] `https://web-of-digital-twins.github.io/`

and literature summarization [22]. AI applications in healthcare can be clustered into predictive models, decision support systems, and big data analytics [23]. Given the critical nature of healthcare, ethical considerations, explainability, and transparency become essential [24, 25].

Given the complexity of healthcare, the integration of MAS provides a natural framework to encapsulate intelligent autonomous behavior to support the automation of healthcare processes. Autonomous agents can model complex processes, where behavior can be programmed, learned, (or *generated* with new approaches [26]). In this work, we focus on programmed agents, particularly Belief-Desire-Intention (BDI) agents [10], which can encode domain and procedural knowledge, making them well-suited for reliable and explainable applications in DT-based healthcare systems.

4.1 Complementarity of Digital Twins and Multi-Agent Systems

Autonomous Agents (AAs) and Digital Twins (DTs) are increasingly adopted in IoT systems to implement intelligent applications in cyber-physical environments. AAs encapsulate goal-driven reasoning and decision-making, closing perception-action loops, and excel at logic inference, adaptive control, learning of control policies, and simulation of complex systems, while DTs digitalize physical entities and can augment their behavior through physics simulation, prediction, and inference.

Across these approaches, several high-level intelligent functionalities emerge – *prediction, simulation, planning, inference, adaptation, and learning* – which can be distributed across AAs and DTs to enhance the capabilities of IoT systems. To clarify the complementary roles of DTs and MAS in engineering intelligent IoT systems, a set of design principles is proposed to guide functional analysis of intelligent functionalities. These principles are derived from the software engineering concept of *separation of concerns* [27] and refined along three dimensions:

- **Specificity:** Measures whether a functionality serves a particular application goal or is generally reusable across multiple goals and applications. General-purpose functionalities, such as predicting asset states or simulating alternate scenarios, are well suited to DTs. Application-specific functionalities, like adaptive control policies or fault diagnosis, are better encapsulated by MAS due to their goal-driven nature.
- **Scoping:** Considers whether a functionality requires information from a single PA or multiple entities across the system. DTs are naturally localized, tied to the assets they digitalize, while MAS can perceive and act across the entire system, supporting global coordination and interaction.
- **Timing:** Evaluates the dependency of a functionality on explicit or implicit notions of time. DTs are typically *time-aware*, maintaining up-to-date representations of their physical twins, whereas MAS are *time-situated*, interacting with the system without necessarily modeling time explicitly.

Using these principles: **MAS** encapsulate application-specific goals, coordinate globally, and operate in a time-situated manner; **DTs** model assets, provide general-purpose services, operate locally, and are inherently time-aware.

This framework provides clear criteria for distributing intelligent functionalities in IoT systems while maintaining flexibility and modularity. The combination of AAs and DTs can give rise to several "micro-architectures" (μ-archs) for integrating AAs and DTs, arising from the application of these principles. The term μ-arch distinguishes these focused integration patterns from traditional top-down reference architectures, adopting a bottom-up approach instead (Figure 4). **(A)** is the most common [27], splitting functionalities between DTs (asset-specific, time-aware) and AAs (application-specific, goal-driven). **(B)** is a DT-only case for low-specificity, local-scope, time-aware functionalities. **(C)** follows the agentification paradigm [28], wrapping assets as agents for localized goals. **(D)** is an antipattern [29], where agents replace functionalities of DTs. Finally, **(E)** augments DTs with internal agents for decision-making or planning, supporting *cognitive* DT architectures [30].

5 Applications in Healthcare

This thesis investigates the application of DTEs in three distinct healthcare domains, demonstrating the versatility and benefits of the proposed framework.

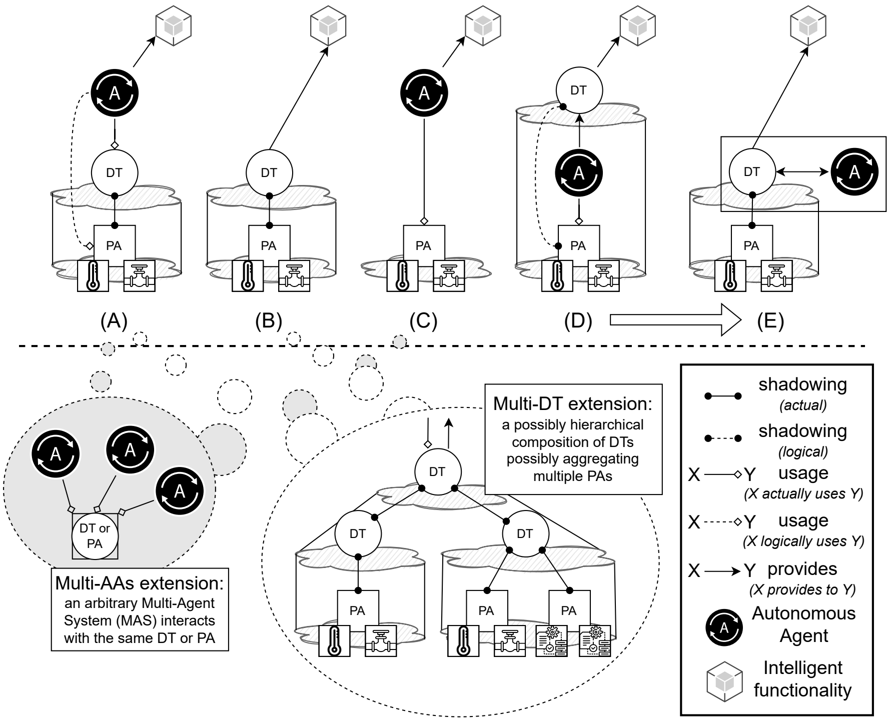


Figure 4: Micro-architectures for the synergistic combination of AAs and DTs. Such architectures can be combined to give shape to the overall system architecture, in a bottom-up way.

Oncological Pharmaceutical Supply Chain Management Pharmaceutical supply chain (PSC) management is critical in oncology due to the high degree of therapy personalization and the severe impact of delays on patient outcomes. The process involves sourcing raw materials, preparing drugs in loco, and administering therapies according to patient-specific schedules. A DTE was proposed to track patients, treatments, production units, and warehouses in real time. By integrating predictive models and monitoring resources, the system may enable optimized warehouse management, right-time logistics, and traceability, supporting the timely preparation and delivery of therapies while improving resource utilization [31].

Operating Room Management Effective management of operating rooms is essential to optimize surgery schedules, improve patient care, and reduce costs. Current practice often relies on manual scheduling and paper-based tracking, which can lead to inefficiencies and lack of accountability. A DTE-based system was implemented to monitor surgeries, patients, operating rooms, and medical staff in real time. The framework collects and standardizes data, detects anomalies, supports dashboard visualization, and integrates simulation models to evaluate scheduling strategies, ultimately enhancing planning, monitoring, and process efficiency [32].

Major Trauma Management Major trauma management is highly time-sensitive, involving emergency call handling, pre-hospital care, and trauma center treatment. Timely coordination of ambulances, rescuers, and trauma teams is crucial for patient outcomes. A DTE ecosystem prototype was developed using heterogeneous DT technologies, integrated through the HWoDT framework. The system tracks missions, ambulances, rescuers, patients, and trauma processes, supports resource allocation, real-time monitoring, and automated documentation, demonstrating interoperability and advanced querying capabilities across heterogeneous DTs.

6 Conclusion

This thesis addressed the engineering challenges of DTs and their integration into DTEs for intelligent healthcare applications. By proposing modular, interoperable architectures and introducing semantic descriptions, the work enables flexible integration and management of heterogeneous DTs. The HWoDT prototype demonstrates how Web standards and semantic technologies can unify diverse DT platforms, supporting scalable and expressive ecosystems. Furthermore, the analysis of MAS clarifies their complementary role with DTs in implementing trustworthy, intelligent applications. The proposed approaches have been validated in healthcare scenarios, highlighting their potential to improve efficiency, interoperability, and decision-making in complex domains. Future work will further explore large-scale deployments, advanced semantic integration, and more complex augmentation functionalities at the ecosystem level, combining the capabilities of multiple DTs.

References

[1] G. L. Tortorella, F. S. Fogliatto, A. Mac Cawley Vergara, R. Vassolo, and R. Sawhney, "Healthcare 4.0: trends, challenges and research directions," *Production Planning & Control*, vol. 31, p. 1245–1260, Nov. 2020.

[2] S. Kraus, F. Schiavone, A. Pluzhnikova, and A. C. Invernizzi, "Digital transformation in healthcare: Analyzing the current state-of-research," *Journal of Business Research*, vol. 123, p. 557–567, Feb. 2021.

[3] M. W. Grieves, *Digital Twins: Past, Present, and Future*, p. 97–121. Cham: Springer International Publishing, 2023.

[4] M. Alazab, L. U. Khan, S. Koppu, S. P. Ramu, I. M, P. Boobalan, T. Baker, P. K. R. Maddikunta, T. R. Gadekallu, and A. Aljuhani, "Digital twins for healthcare 4.0—recent advances, architecture, and open challenges," *IEEE Consumer Electronics Magazine*, vol. 12, p. 29–37, Nov. 2023.

[5] A. Ricci, A. Croatti, S. Mariani, S. Montagna, and M. Picone, "Web of digital twins," *ACM Trans. Internet Technol.*, vol. 22, nov 2022.

[6] R. T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures.* Phd dissertation, University of California, Irvine, 2000.

[7] K. Toumura, M. Lagally, M. McCool, and R. Matsukura, "Web of things (wot) architecture 1.1," W3C recommendation, W3C, Dec. 2023. https://www.w3.org/TR/2023/REC-wot-architecture11-20231205/.

[8] A. Giulianelli, S. Burattini, A. Ciortea, and A. Ricci, "Hypermedia-based digital twin ecosystems: A proposal for digital twin interoperability." *Accepted*, 2025.

[9] M. J. Wooldridge, *An Introduction to MultiAgent Systems, Second Edition.* Wiley, 2009.

[10] M. Bratman, *Intention, Plans, and Practical Reason.* Cambridge: Cambridge, MA: Harvard University Press, 1987.

[11] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models," *Proceedings of the IEEE*, vol. 108, p. 1785–1824, Oct. 2020.

[12] F. Tao, M. Zhang, Y. Liu, and A. Nee, "Digital twin driven prognostics and health management for complex equipment," *CIRP Annals*, vol. 67, no. 1, pp. 169–172, 2018.

[13] P. Bellavista, N. Bicocchi, M. Fogli, C. Giannelli, M. Mamei, and M. Picone, "Requirements and design patterns for adaptive, autonomous, and context-aware digital twins in industry 4.0 digital factories," *Computers in Industry*, vol. 149, p. 103918, Aug. 2023.

[14] M. Picone, M. Mamei, and F. Zambonelli, "WLDT: A general purpose library to build iot digital twins," *SoftwareX*, vol. 13, p. 100661, 2021.

[15] ETSI Specialist Task Forces (STF) 628, "SmartM2M; digital twins communication requirements," TS TS-103-845-V1.1.1, European Telecommunications Standards Institute (ETSI), February 2024.

[16] E. Ferko, A. Bucaioni, and M. Behnam, "Architecting digital twins," *IEEE Access*, vol. 10, pp. 50335–50350, 2022.

[17] C. E. B. López, "Real-time event-based platform for the development of digital twin applications," *The International Journal of Advanced Manufacturing Technology*, vol. 116, p. 835–845, June 2021.

[18] S. Acharya, A. A. Khan, and T. Päivärinta, "Interoperability levels and challenges of digital twins in cyber–physical systems," *Journal of Industrial Information Integration*, vol. 42, p. 100714, Nov. 2024.

[19] Y. Peng, M. Zhang, F. Yu, J. Xu, and S. Gao, "Digital twin hospital buildings: An exemplary case study through continuous lifecycle integration," *Advances in Civil Engineering*, vol. 2020, no. 1, p. 8846667, 2020.

[20] A. Giulianelli, S. Burattini, A. Ciortea, and A. Ricci, "Hwodt framework: A toolchain to build interoperable digital twin ecosystems," *SoftwareX*, vol. 31, p. 102275, 2025.

[21] K.-H. Yu, A. L. Beam, and I. S. Kohane, "Artificial intelligence in healthcare," *Nature Biomedical Engineering*, vol. 2, p. 719–731, Oct. 2018.

[22] S. Sai, A. Gaur, R. Sai, V. Chamola, M. Guizani, and J. J. P. C. Rodrigues, "Generative AI for transformative healthcare: A comprehensive study of emerging models, applications, case studies, and limitations," *IEEE Access*, vol. 12, p. 31078–31106, 2024.

[23] S. Secinaro, D. Calandra, A. Secinaro, V. Muthurangu, and P. Biancone, "The role of artificial intelligence in healthcare: a structured literature review," *BMC Medical Informatics and Decision Making*, vol. 21, p. 125, Apr. 2021.

[24] S. Bharati, M. R. H. Mondal, and P. Podder, "A review on explainable artificial intelligence for healthcare: Why, how, and when?," *IEEE Transactions on Artificial Intelligence*, vol. 5, p. 1429–1442, Apr. 2024.

[25] M. Alonso, A. M. Astobiza, and R. Ortega Lozano, "AI-mediated healthcare and trust. a trust-construct and trust-factor framework for empirical research," *Artificial Intelligence Review*, vol. 58, p. 337, Aug. 2025.

[26] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic AI: Autonomous intelligence for complex goals—a comprehensive survey," *IEEE Access*, vol. 13, p. 18912–18936, 2025.

[27] S. Mariani, M. Picone, and A. Ricci, "About digital twins, agents, and multiagent systems: A cross-fertilisation journey," in *Autonomous Agents and Multiagent Systems. Best and Visionary Papers* (F. S. Melo and F. Fang, eds.), (Cham), p. 114–129, Springer International Publishing, 2022.

[28] P. Pico-Valencia and J. A. Holgado-Terriza, "Agentification of the internet of things: A systematic literature review," vol. 14, no. 10.

[29] H. Wan, M. David, and W. Derigent, "Modelling digital twins as a recursive multi-agent architecture: application to energy management of communicating materials," *IFAC-PapersOnLine*, vol. 54, p. 880–885, Jan. 2021.

[30] R. Minerva, N. Crespi, R. Farahbakhsh, and F. M. Awan, *Artificial Intelligence and the Digital Twin: An Essential Combination*, p. 299–336. Cham: Springer International Publishing, 2023.

[31] S. Burattini, S. Montagna, N. Gentili, F. Galardi, R. Vespignani, P. Zanatto, A. Croatti, and A. Ricci, "A digital twins approach for oncologic pharmaceutical supply chain," in *IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom 2024 - Workshops, Biarritz, France, March 11-15, 2024*, pp. 94–99, IEEE, 2024.

[32] S. Burattini, S. Montagna, A. Croatti, N. Gentili, A. Ricci, L. Leonardi, S. Pandolfini, and S. Tosi, "An ecosystem of digital twins for operating room management," in *36th IEEE International Symposium on Computer-Based Medical Systems, CBMS 2023, L'Aquila, Italy, June 22-24, 2023* (J. R. Almeida, M. Spiliopoulou, J. A. Benítez-Andrades, G. Placidi, A. R. González, R. Sicilia, and B. Kane, eds.), pp. 770–775, IEEE, 2023.