Digital Twin: Graph Formulations for Managing Complexity and Uncertainty

Luwen Huang
Department of Computer Science
University of Texas at Austin
Email: luwen@cs.utexas.edu

Michael Kapteyn and Karen E. Willcox
Oden Institute for Computational Engineering and Sciences
University of Texas at Austin
Email: mkapteyn@utexas.edu, kwillcox@oden.utexas.edu

***Abstract*—This paper articulates the importance of graphical models and graph-based methods as fundamental enablers of Digital Twins. Graph-based representations are well known to be suited for describing complex systems where the connections between entities are as important as the entities themselves. The interconnections within and across data, models, and decisions are central to a Digital Twin's value. Not only does a graph emphasize the scalable representation of such interrelationships, it also provides a natural mathematical setting for addressing uncertainty and complexity—arguably the two biggest barriers to scalable deployment and adoption of Digital Twins. We discuss how recent advances in theory and algorithms for large-scale knowledge graphs and graphical models can be combined in a multi-layered formulation to provide a powerful foundation for achieving scalable Digital Twins for complex systems.**

***Index Terms*—Digital twin, knowledge graph, ontology, probabilistic graphical model**

1. Introduction

A Digital Twin is "a set of virtual information constructs that mimics the structure, context, and behavior of a natural, engineered, or social system (or system-of-systems), is dynamically updated with data from its physical twin, has a predictive capability, and informs decisions that realize value." [2] The National Academies report [2] emphasizes that a Digital Twin goes beyond being just simulation and modeling, highlighting that "the bidirectional interaction between the virtual and the physical is central to the digital twin." The value of Digital Twins for continuous monitoring, predictive maintenance or intervention, and improved decision-making has been widely recognized [3], [4], [5] across diverse fields including precision healthcare [6], [7], sustainable energy [8], and aerospace engineering [9]. Despite this promise, scaling digital twins up to the level of complexity, reliability, and computational efficiency required for real-world applications remains challenging [10].

A Digital Twin is highly interconnected, integrating data, models, and decisions across the physical and virtual worlds (depicted in Figure 1). To reflect this interconnectedness in the mathematical and computational underpinnings of a Digital Twin, it is essential to be able to represent *relationships* in a scalable way. This paper makes the case for graphical models and graph-based methods as fundamental enablers of Digital Twins.

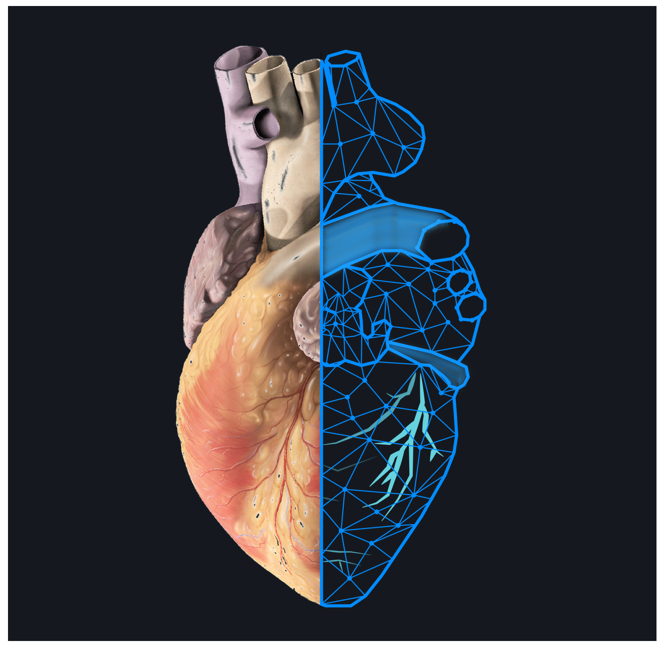

Figure 1. A Digital Twin of a human heart. Left side of the image is adapted from [1].

A graph is a mathematical structure consisting of a set of entities and their relationships. Entities are represented as vertices in the graph and the relationships between pairs of entities are represented as edges between vertices. Edges encode directionality and can be directed, as seen in the two pairs of edges $B \rightarrow C$ and $D \rightarrow E$ in Figure 2, or they can be undirected, as in the edge between A and B and the edge between C and D.

A key differentiating factor in a graph is that the graph structure allows for the representation of edges as first-class citizens [11]. That is, the edges are objects in their own right that can be named, passed around in functions, and otherwise manipulated [12]. Because of this explicit representation of relationships, graphs are naturally suited for describing systems where the connections between entities

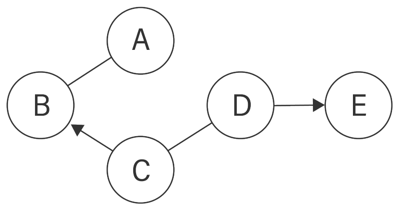

Figure 2. A graph with five vertices and four edges.

are as important as the entities themselves. This makes a graph a natural mathematical foundation for a Digital Twin.

From Jordan's book *Learning in Graphical Models* [13]:

> "Graphical models, a marriage between probability theory and graph theory, provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering—uncertainty and complexity... Fundamental to the idea of a graphical model is the notion of modularity: a complex system is built by combining simpler parts. Probability theory serves as the glue whereby the parts are combined, ensuring that the system as a whole is consistent and providing ways to interface models to data. Graph theory provides both an intuitively appealing interface by which humans can model highly interacting sets of variables and a data structure that lends itself naturally to the design of efficient general-purpose algorithms."

This quote highlights the potential value of formulating a Digital Twin using graphical models. Not only does a graphical model emphasize the modeling of interrelationships, which is critical in the Digital Twin setting, it also provides a natural mathematical setting for addressing uncertainty and complexity—arguably the two biggest barriers to scalable deployment and adoption of Digital Twins. Furthermore, recent years have seen significant advances in theory and algorithms for large-scale graphical models, advances that could be leveraged in achieving scalable Digital Twins for complex systems.

Using a graph foundation, this paper lays out a multi-layered formulation for Digital Twins of complex systems. In Section 2, we describe a Digital Twin foundational layer, built on knowledge graphs. In Section 3, we describe a Digital Twin predictive layer, built on probabilistic graphical models. In Section 4, we highlight how these graph-based formulations are being deployed in scalable computational Digital Twin technologies.

2. The Foundational Layer

A Digital Twin encodes knowledge that, for a complex system, spans many aspects of the system's structure, context, and behavior. In a Digital Twin of a cancer patient, for example, this knowledge may encompass data from clinical observational records, data from quantitative clinical assessments, predictive mathematical models that encode biological principles governing tumor progression and response to treatment, statistical models that encode relationships between genomic factors and cancer progression, and much more. The knowledge domain of Digital Twins is characterized by (1) its diversity, heterogeneity, and multimodal nature, (2) its complex interconnections, and (3) the need for knowledge integrity. Without a structured approach to managing these complex data, building a functional Digital Twin would be impossible. Conversely, a semantic, integrated knowledge structure that ensures knowledge integrity is the core enabler of Digital Twin functions. This structure forms what we call the foundational layer of the Digital Twin, as illustrated in Figure 3.

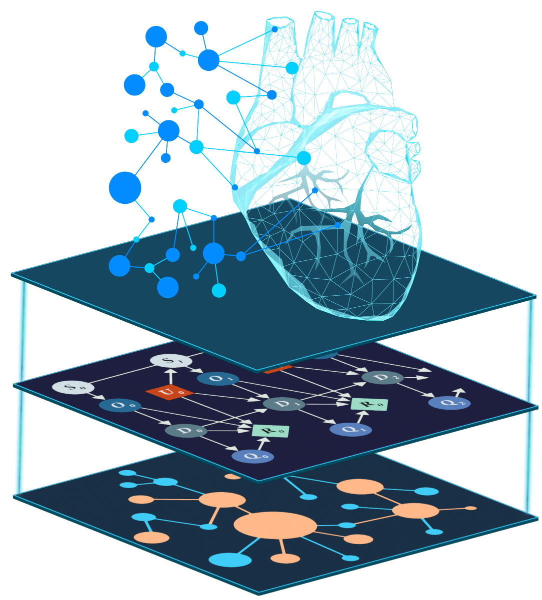

Figure 3. With graphs as the foundation, a multi-layered formulation for Digital Twins consists of the foundational layer, the predictive layer, and applications that build upon these layers.

2.1. Challenges, needs, and the role of the foundational layer

Knowledge diversity. Digital Twins process a vast array of data that varies widely in source, format, and subject matter [14], [15]. Different pieces of data also differ in how, when and at what level of detail they are acquired. For example, in an unmanned aerial vehicle (UAV) Digital Twin [16], sensor data for temperature, wind speed, and structural strain are collected and transmitted at different rates and different levels of fidelity. In addition to sensor data, the Digital Twin incorporates other diverse types of knowledge, such as inspection logs and flight logs. Even though these data differ in origin, format, and timing, the Digital Twin must manage this diversity and integrate the data in a cohesive way. To do so, there is a need for knowledge semantics — an articulation of how knowledge is structured, what are the atoms of knowledge, how these atoms are related to each other, and how they comprise the knowledge domain.

Dense interconnectivity. Knowledge components within a Digital Twin are highly interconnected. This interconnectivity arises from the need to map the digital representation to the physical world. These mappings create connections between different parts of the knowledge domain, linking data and models within the Digital Twin. Additionally, the complexity of real-world systems contributes to this high level of connectivity [17]. Real-world systems often consist of numerous interdependent components that interact closely. For example, in a Digital Twin for diabetes management, [18] integrates the dynamic relationships among diverse data sources such as insulin levels, genetics, and dietary habits.

Knowledge Integrity. In a Digital Twin, bidirectional data flow is a key principle. This means that operations are not limited to querying knowledge; they also involve updating knowledge [19]. These operations can introduce errors, ranging from computational inaccuracies to logical inconsistencies, such as violations of ontological rules. Such errors can result in failure or remain hidden and be carried forward, impacting future operations. Because a Digital Twin is an evolving, dynamically-updating construct, typically applied in high-stakes applications, maintaining knowledge integrity is crucial.

These characteristics highlight the need to represent, organize, and engineer knowledge in Digital Twins. The resulting knowledge structure must provide clear, structured semantics that are robust to system dynamics and easy to understand from a human perspective. While knowledge engineering is a well-established field, it is only in the last decade, with the proliferation of Digital Twins, that significant attention has been given to structuring knowledge specifically within this context. Knowledge graphs have emerged as the leading approach for structuring this foundational layer [15], [20], [21], [22], [23], [24].

2.2. The Ontology and Knowledge Graph

Key in knowledge structuring of Digital Twins is the *ontology*. An ontology is a generalized semantic data model that defines the types of entities, the relationships between them, the attributes attached to these entities, and the rules governing them [25]. For example, Figure 4 illustrates an ontology for a UAV Digital Twin that specifies the types, attributes, and rules for entities and their relationships. The ontology defines five types of entities residing in the physical world: Component, Sensor, Measurement, Physical state parameter, and Control action. Additionally, it defines three types of entities in the digital world: Model, Digital state parameter, and Quantity of interest. The ontology also establishes nine types of relationships to describe how these entities interact. For instance, a Sensor *generates* a Measurement.

Moreover, the ontology specifies attributes and rules for entities and relationships. For example, a Sensor has a

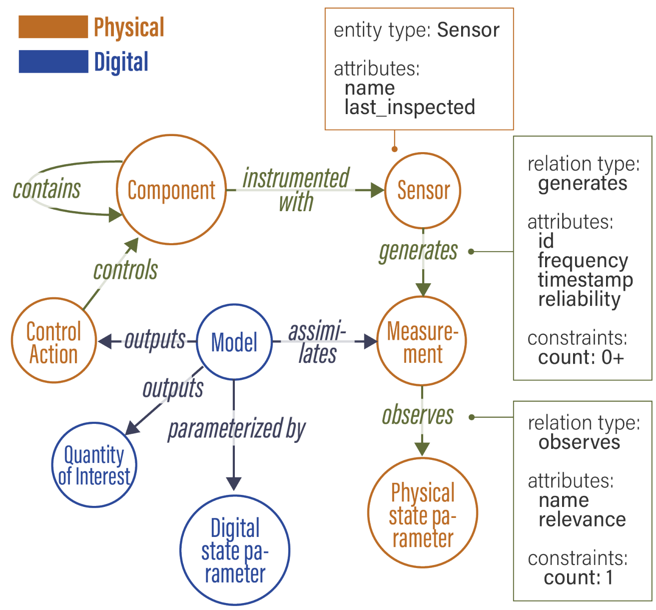

Figure 4. An ontology for an unmanned aerial vehicle Digital Twin that specifies the entities, relationships between entities and rules for entities and relationships.

`name` and a `last_inspected` timestamp. Crucially, relationships, being first-class entities in a graph structure, also have attributes. The *generates* relationship has the attributes `frequency`, which describes how often measurements are generated, and `reliability`, which indicates how reliable the generation process has been over a recent time period. Ontologies also enforce rules regarding the nature of the data [26]. For instance, the *generates* relationship can have zero to infinitely many edges, representing scenarios from a sensor being inoperational (zero times) to a non-terminating physical process (infinite times). Conversely, the *observes* relationship mandates exactly one edge between a Measurement and a Physical state parameter, ensuring that each measurement corresponds to one and only one physical state parameter.

A *knowledge graph* is the instantiation and application of an ontology to specific entities. While an ontology provides an abstract framework of rules, axioms, and semantics, a knowledge graph represents a concrete realization of this framework, applied to actual data. It contains the entities and relationships specific to the data domain. Figure 5 shows an illustrative knowledge graph that results from applying the abstract ontology presented in Figure 4. By applying ontological design to data, the resulting knowledge graph delivers powerful semantics, offering clear naming and meaning of entities and relationships that enable both user-friendly interaction and formal reasoning about the data. The graph structure is easily understood by viewers, as it visually clarifies what entities exist within the data and how these entities relate to each other, thanks to the intuitive directionality and verb-based naming of edges. For example, in the graph depicted in Figure 5, a viewer

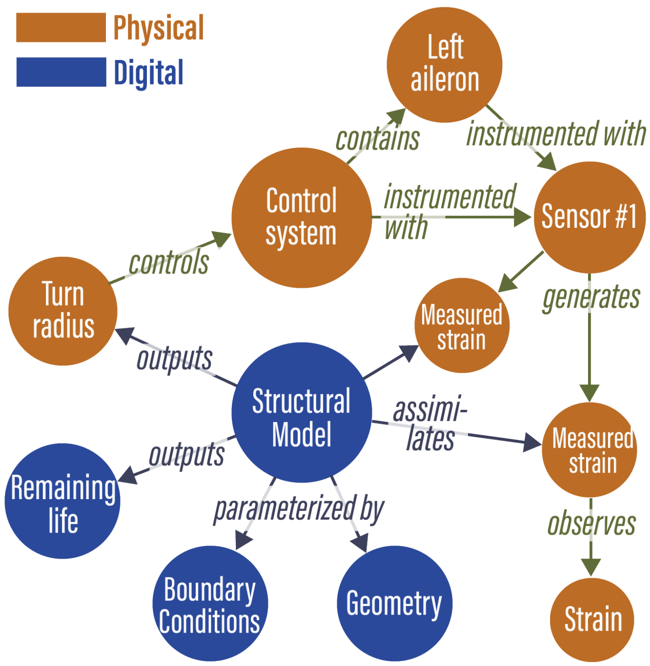

Figure 5. Notional example of a knowledge graph instantiated by applying the ontology in Figure 4 to the unmanned aerial vehicle domain.

naturally grasps the one-way nature of the *parameterized by* edge — the STRUCTURAL MODEL is parameterized by BOUNDARY CONDITIONS and GEOMETRY, but the latter are not "parameterized by" models. While this may seem trivial in this example, data intuitiveness should not be discounted in real-life Digital Twins, where knowledge is complex and relationships between elements may not be immediately obvious. This intuitiveness is particularly valuable when working with the data, such as in querying and developing code.

The formal semantics embedded in a knowledge graph play a crucial role in ensuring the correctness of the system. These semantics enable the application of mathematical techniques, ensuring that the relationships and operations within the graph adhere to predefined ontological rules and constraints [27]. For example, the Depth-First-Search (DFS) algorithm can traverse the graph and check that no cycles are formed with relationships of a certain type. This is mathematically formalized as follows: let $G = (V, E)$ denote a directed graph with the set of vertices V and edges $E = V \times V$. A cycle is defined as the sequence of vertices $v_1 \ldots v_n \in V$ such that $(v_1, v_2) \ldots (v_{n-1}, v_n), (v_n, v_1) \in E$. Annotating with type semantics, we can specify a cycle with a set of types $T_1 \ldots T_k \in \Sigma$, where Σ is the type alphabet. Such a typed cycle can be denoted as $(\Gamma : T_1, v_1, v_2) \ldots ((\Gamma : T_{n-1}, v_{n-1}, v_n), ((\Gamma : T_n, v_n, v_1)$. As a concrete example that builds upon the knowledge graph shown in Figure 5, to discover cycles where a COMPONENT *contains* itself (cyclic loop), we write the typed cycle $(\Gamma : contains, v_1, v_2) \ldots ((\Gamma : contains, v_{n-1}, v_n), ((\Gamma : contains, v_n, v_1)$ where $v_1 \ldots v_n$ is of type COMPONENT. Applied to the graph, the DFS algorithm discovers the violating cycle shown in Figure 6. Formal semantics make it easy to formally specify and verify node-level, edge-level and graph-level properties, which is crucial in real-world Digital Twin knowledge graphs.

In addition to correctness, formal semantics also contribute to scalable transactions in bidirectional data flows. Graphs have the index-free adjacency property, meaning that each node in the graph directly points to its adjacent nodes without the need for an index lookup [28]. This allows for rapid traversal and transformation of the graph structure. When combined with semantics, querying and updating the knowledge graph can be further optimized with techniques such as typed indexing and query rewriting.

The knowledge graph structure also supports knowledge representation at multiple levels of granularity, which is a common requirement in Digital Twins. For example, in a UAV Digital Twin, low-level granular sensor observations are regularly updated into the knowledge graph. However, decision-makers might need to query information at a less-granular level, such as the sequence of maneuvers executed by the UAV. This less-granular level might not be explicitly represented in the foundational knowledge graph as distinct entities and relationships. Graph transformations address this need by rolling up (or unrolling) entities and relationships of the knowledge graph. In a graph transformation, an input graph is transformed into an output graph through a series of mathematical operations. These transformations enable the creation of new knowledge graphs, which are distinct and mathematically-derived from the original knowledge graph. Graph transformations can be highly scalable due to the index-free adjacency property of graphs.

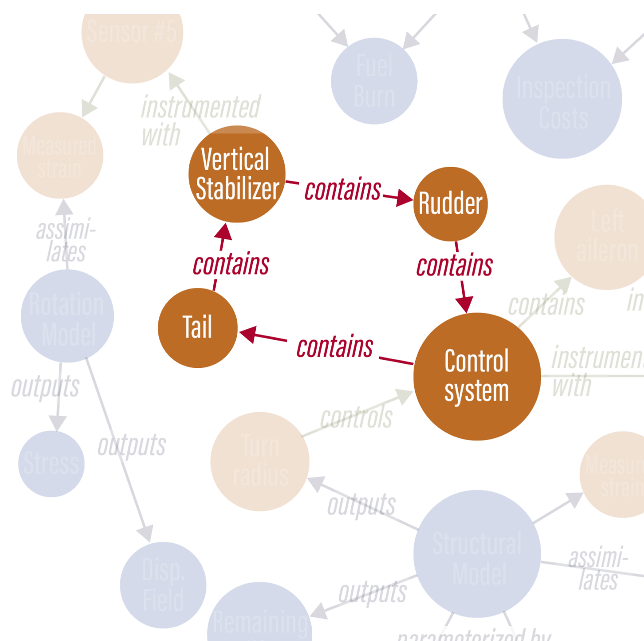

Figure 6. An illegal cycle, formalized as a mathematical statement annotated with type semantics, discovered by depth-first-search traversal through the graph.

3. The Predictive Layer

A core tenet of the Digital Twin is its predictive capability, which is essential for informing critical decisions [2], [10]. Predictive capability requires the Digital Twin not only to issue predictions beyond the available data, but also to quantify uncertainty and levels of confidence associated with those predictions. Further, the Digital Twin requires the ability to update itself and its predictions as the physical world evolves. Without this dynamic updating, predictions become stale and lose relevance. Maintaining predictive capability is challenging due to the inherent uncertainty in the system [29]. This uncertainty can manifest in various forms. For instance, in a Digital Twin that supports UAV operation or UAV manufacturing, there may be uncertainty regarding the material properties of a component. Similarly, in a Digital Twin designed to optimize radiation therapy for a cancer patient, there may be uncertainty in the model parameters used for tumor characterization [30]. The Digital Twin predictive layer must capture this uncertainty, reason about it, and propagate it into the future as the Digital Twin system evolves.

Uncertainty is often modeled using random variables, which assign probabilities to different possible outcomes. For example, consider a digital twin of a cancer patient. We can define a random variable O that represents the MRI observational data and the imaging data's associated uncertainties. We can define another random variable D that represents the digital representation of the patient's tumor state, inferred from the MRI observational data, and also uncertain. Similarly, we can introduce a random variable U that represents a decision action around radiation dosage, and a random variable R that represents a reward associated with patient outcomes. Because of the dense interconnectivity of elements in a Digital Twin (see Section 2), the random variables representing the status of these elements are rarely independent; indeed, the patient outcomes depend on the state of the tumor and the radiation dosage levels.

In real-world Digital Twin applications there may be millions of random variables with complex dependencies, so sophisticated methods are required to model and represent these elements. State-of-the-art methods for such problems are typically based on graphical structures. One prominent model used in Digital Twins is the graphical model. Figure 7 depicts a *probabilistic graphical model* (PGM) for the illustrative cancer patient example. The PGM uses a graph structure to express the conditional dependencies between random variables. Random variables are represented as nodes in the graph and edges between them express conditional probabilities.

In this example the patient outcomes, R, depend on both the tumor state D and the radiation dosage U, as represented by the directed edges in Figure 7. PGMs are powerful tools for modeling uncertainty, making predictions, and governing state changes in complex systems because they admit scalable inference algorithms. For example, if we are able to obtain updated imaging data then we can algorithmically refine our estimate of the tumor state, and in turn refine our estimate of an optimized radiation dosage targeted to improve patient outcomes.

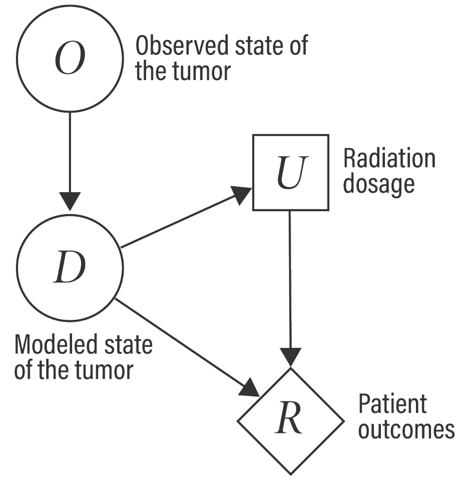

Figure 7. Probabilistic graphical model formulation of a Digital Twin representing the interdependencies between MRI data, modeled state of the tumor, radiation dosage, and patient outcomes.

A PGM formulation of a Digital Twin therefore provides the flexible mathematical framework to enable data-driven asset monitoring, digital twin model updating, model-based prediction, planning, and optimal control [16].

The Digital Twin PGM framework proposed in [16], formulates a PGM for the Digital Twin system to govern how system components evolve over time. This approach formulates a mathematical abstraction of the system and define six key elements as illustrated in Figure 8. These elements comprise representations of quantities in the physical world and the virtual world: (1) the partially observable physical state, S; (2) the physical observational data O; (3) the physical control inputs U that influence the asset state; (4) the virtual digital twin state, D; (5) quantities of interest Q predicted by the Digital Twin's virtual models; and (6) a reward R that evaluates costs and performance of the system. Here, the subscript t denotes a discrete time step.

Once these elements are defined, one formulates the PGM describing the system. For example, Figure 9 shows the PGM for a UAV Digital Twin with the six previously defined elements S, O, U, D, Q, and R. The figure illustrates three timesteps at $t = 0$ (inception), $t = t_c$ (present), and $t = t_p$ (future) to show the conditional dependencies between and among components as the system evolves over time. For instance, from the PGM, one reads that the conditional dependency of the digital state at the present time D_p is dependent on the initial digital state D_0 and the data O_p, as well as the previous action U_0. In this way, the PGM framework enables probabilistic inference over the graph, such as predicting into the future, analyzing over the past, or tracking current conditions. The Digital Twin PGM framework is general and can be applied to various Digital Twins; for instance, see [16] for its application to UAV dynamic mission replanning, [31] for its application

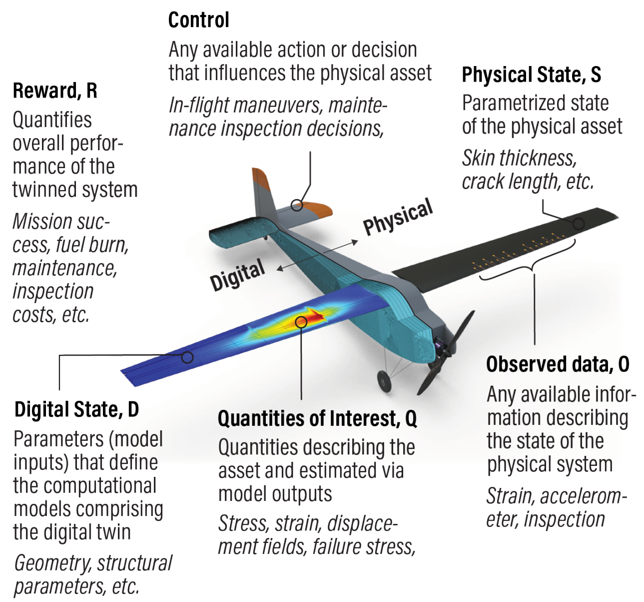

Figure 8. The six elements that define a probabilistic graphical model of a Digital Twin, with examples for an unmanned aerial vehicle. Figure modified from [16].

in civil engineering structures, and [30] for its application in cancer patient Digital Twins.

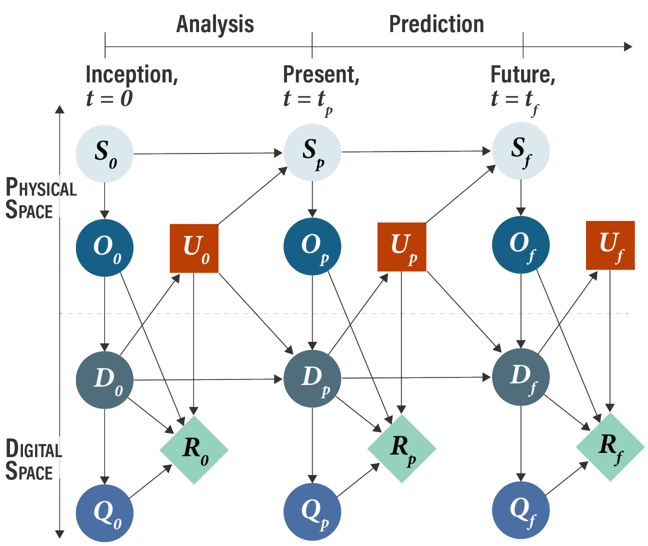

Figure 9. A probabilistic graphical model that describes the dynamic evolution and interdependencies of the six elements of a Digital Twin. Figure modified from [16].

Although both PGMs and knowledge graphs are graphical structures, it is important not to conflate the two. In a PGM, the vertices represent random variables, and the edges between them represent dependencies between these variables. In contrast, a knowledge graph has vertices that represent individual data elements, and edges that signify permissible relationships between these data elements. The two graphical methods are complementary: the knowledge graph serves as a foundational layer that explicitly supports rich semantics, which embeds key information about the digital twin; the PGM brings in the formal quantification of uncertainty, which is critical for the use of digital twins in predictive settings.

4. Graph-based Digital Twin Implementation Technologies

Implementation strategies for graph-based methods underpinning Digital Twins vary across different domains. Here we highlight some examples of Digital Twin implementations that build on graph-based approaches (noting that this is not a comprehensive list). These examples highlight the power of graph-based approaches in providing scalable and flexible implementations for Digital Twins.

In the literature, the graph database Neo4j is frequently cited as a preferred solution for implementing knowledge graphs. Examples that employ Neo4j for Digital Twin knowledge graphs include [23] and [32]. In another example, a study from IBM describes their in-house efforts to build a Digital Twin platform [24]. This work builds a knowledge graph that provides a layer of semantic data access and guaranteed data consistency in bidirectional data flow. Microsoft Azure Digital Twins employs a “Digital Twin Graph” that expresses relationships between Digital Twin instances. This graph can be visualized and queried via `SQL`. Similarly, Amazon IoT TwinMaker employs a knowledge graph to express relationships between entities and their components. This graph can be queried via the `PartiQL` query language. Knowledge graphs are also used as the underpinning for a Digital Twin architecture proposed in [33], with a software prototype illustrated for examples in manufacturing and underwater ship inspections.

The recent literature also contains a number of examples of PGM-based implementations of Digital Twins, where there is an emphasis on Bayesian methods to quantify uncertainty in Digital Twin estimates and predictions. Examples of these PGM-based implementations include aircraft structural health monitoring [34], manufacturing systems [35], spacecraft structural health monitoring [36], and space habitats [37]. The PGM-based approach summarized in Section 3 and a reference implementation is provided in [16]; this implementation is also employed in [30] and [31].

5. Discussion

We have described knowledge graphs and PGMs, and why they are central enablers of Digital Twins. This paper focuses on a mathematical approach to structuring Digital Twins. From a computing perspective, managing bidirectional data flow requires the consideration of program execution, event timing, and correctness (i.e., components being error-free). In many applications of Digital Twins, this requires the handling of distributed components, which are systems spread across multiple locations or devices that must work together. In real-world systems, the timing of data events is often indeterministic due to network delays, variable processing times, and other unpredictable factors. Ensuring correct program execution under these conditions is one of the computing challenges in Digital Twins, though it is beyond the scope of this paper.

Acknowledgments

The authors acknowledge support from Department of Energy grants DE-SC0021239 and DE-SC002317, and Air Force Office of Scientific Research grants FA9550-21-1-0084, FA9550-22-1-0419, and A24-0380-S002.

References

[1] OpenStax Anatomy and Physiology, "Human heart photo," 2016, this work is licensed under the Creative Commons Attribution 4.0 International License. "Human heart" by teynOpenStax Anatomy and Physiology is licensed under CC by 4.0. / Cropped with illustration added to original. [Online]. Available: https://commons.wikimedia.org/wiki/File:2000_Human_Heart_Photo.jpg

[2] National Academies of Science, Engineering and Medicine, *Foundational Research Gaps and Future Directions for Digital Twins*. Washington, DC: The National Academies Press, 2024. [Online]. Available: https://nap.nationalacademies.org/catalog/26894/foundational-research-gaps-and-future-directions-for-digital-twins

[3] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, pp. 85–113, 2017.

[4] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020.

[5] M. W. Grieves, "Digital twins: past, present, and future," in *The Digital Twin*. Springer, 2023, pp. 97–121.

[6] T. Hernandez-Boussard, P. Macklin, E. J. Greenspan, A. L. Gryshuk, E. Stahlberg, T. Syeda-Mahmood, and I. Shmulevich, "Digital twins for predictive oncology will be a paradigm shift for precision cancer care," *Nature Medicine*, vol. 27, no. 12, pp. 2065–2066, 2021.

[7] R. Laubenbacher, B. Mehrad, I. Shmulevich, and N. Trayanova, "Digital twins in medicine," *Nature Computational Science*, vol. 4, no. 3, pp. 184–191, 2024.

[8] F. Stadtmann, A. Rasheed, T. Kvamsdal, K. A. Johannessen, O. San, K. Kölle, J. O. Tande, I. Barstad, A. Benhamou, T. Brathaug *et al.*, "Digital twins in wind energy: Emerging technologies and industry-informed future directions," *IEEE Access*, 2023.

[9] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US Air Force vehicles," in *Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2012, pp. AIAA Paper 2012–1818.

[10] S. A. Niederer, M. S. Sacks, M. Girolami, and K. Willcox, "Scaling digital twins from the artisanal to the industrial," *Nature Computational Science*, vol. 1, no. 5, pp. 313–320, 2021.

[11] J. Pokorný, "Functional querying in graph databases," *Vietnam Journal of Computer Science*, vol. 5, no. 2, pp. 95–105, 2018.

[12] M. Scott, *Programming Language Pragmatics*. Morgan Kaufmann Publishers, 2006.

[13] M. I. Jordan, *Learning in Graphical Models*. MIT press, 1999.

[14] B. Meyers, J. Van Noten, P. Lietaert, B. Tielemans, H. Hristov, D. Maes, and K. Gadeyne, "Knowledge graphs in digital twins for manufacturing — Lessons learned from an industrial case at Atlas Copco Airpower," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 13–18, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896322016263

[15] D. Peters and S. Schindler, "FAIR for digital twins," *CEAS Space Journal*, vol. 16, no. 3, pp. 367–374, 2024.

[16] M. G. Kapteyn, J. V. R. Pretorius, and K. E. Willcox, "A probabilistic graphical model foundation for enabling predictive digital twins at scale," *Nature Computational Science*, vol. 1, no. 5, pp. 337–347, 2021.

[17] L. Mollica, C. Leli, F. Sottotetti, S. Quaglini, L. Locati, and S. Marceglia, "Digital twins: A new paradigm in oncology in the era of big data," *ESMO Real World Data and Digital Oncology*, 2024.

[18] F. Rad, R. Hendawi, X. Yang, and J. . Li, "Personalized diabetes management with digital twins: A patient-centric knowledge graph approach," *Journal of Personalized Medicine*, vol. 14, no. 4, 2024.

[19] A. Ferrari and K. Willcox, "Digital twins in mechanical and aerospace engineering," *Nature Computational Science*, vol. 4, no. 3, pp. 178–183, 2024.

[20] J. Akroyd, S. Mosbach, A. Bhave, and M. Kraft, "Universal digital twin — A dynamic knowledge graph," *Data-Centric Engineering*, vol. 2, p. e14, 2021.

[21] E. Kharlamov, F. Martin-Recuerda, B. Perry, D. Cameron, R. Fjellheim, and A. Waaler, "Towards semantically enhanced digital twins," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 4189–4193.

[22] E. G. Kalaycı, I. Grangel González, F. Lösch, G. Xiao, A. ul Mehdi, E. Kharlamov, and D. Calvanese, "Semantic integration of Bosch manufacturing data using virtual knowledge graphs," in *The Semantic Web – ISWC 2020*. Cham: Springer International Publishing, 2020, pp. 464–481.

[23] E. Karabulut, S. F. Pileggi, P. Groth, and V. Degeler, "Ontologies in digital twins: A systematic literature review," *Future Generation Computer Systems*, vol. 153, pp. 442–456, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X23004739

[24] J. Ploennigs, K. Semertzidis, F. Lorenzi, and N. Mihindukulasooriya, "Scaling knowledge graphs for automating AI of digital twins," in *The Semantic Web – ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings*, 2022, p. 810–826.

[25] T. Gruber, *Ontology*. Boston, MA: Springer US, 2009, pp. 1963–1965. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_1318

[26] G. Cima, A. Poggi, and M. Lenzerini, "The notion of abstraction in ontology-based data management," *Artificial Intelligence*, vol. 323, p. 103976, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370223001224

[27] J. Hidders, "Typing graph-manipulation operations," in *Database Theory — ICDT 2003*, D. Calvanese, M. Lenzerini, and R. Motwani, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 394–409.

[28] J. Pokorný, "Graph databases: Their power and limitations," in *Computer Information Systems and Industrial Management*, K. Saeed and W. Homenda, Eds. Cham: Springer International Publishing, 2015, pp. 58–69.

[29] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu, "A comprehensive review of digital twin—Part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives," *Structural and Multidisciplinary Optimization*, vol. 66, no. 1, p. 1, 2022.

[30] A. Chaudhuri, G. Pash, D. A. n. Hormuth, G. Lorenzo, M. Kapteyn, C. Wu, E. A. B. F. Lima, T. E. Yankeelov, and K. Willcox, "Predictive digital twin for optimizing patient-specific radiotherapy regimens under uncertainty in high-grade gliomas." *Frontiers in Artificial Intelligence*, 2023.

[31] M. Torzoni, M. Tezzele, S. Mariani, A. Manzoni, and K. E. Willcox, "A digital twin framework for civil engineering structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 418, p. 116584, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782523007089

[32] C. Ramonell, R. Chacón, and H. Posada, "Knowledge graph-based data integration system for digital twins of built assets," *Automation in Construction*, vol. 156, p. 105109, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580523003692

[33] M. Waszak, A. N. Lam, V. Hoffmann, B. Elvesæter, M. F. Mogos, and D. Roman, "Let the asset decide: Digital twins with knowledge graphs," in *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*, 2022, pp. 35–39.

[34] C. Li, S. Mahadevan, Y. Ling, S. Choze, and L. Wang, "Dynamic Bayesian network for aircraft wing health monitoring digital twin," *AIAA Journal*, vol. 55, no. 3, pp. 930–941, 2017.

[35] A. K. Ghosh, A. S. Ullah, and A. Kubo, "Hidden Markov model-based digital twin construction for futuristic manufacturing systems," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 33, no. 3, pp. 317–331, 2019.

[36] Y. Ye, Q. Yang, F. Yang, Y. Huo, and S. Meng, "Digital twin for the structural health management of reusable spacecraft: A case study," *Engineering Fracture Mechanics*, vol. 234, p. 107076, 2020.

[37] N. Gratius, Y. Hou, M. Bergés, and B. Akinci, "Lessons learned on the implementation of probabilistic graphical model-based digital twins: A space habitat study," *Journal of Space Safety Engineering*, vol. 10, no. 2, pp. 172–181, 2023.