Requirements for an Ontology of Digital Twins

Claudio Barros[1], Rebecca Salles[2], Eduardo Ogasawara[2],
Giancarlo Guizzardi[3,4], and Fabio Porto[1]

[1] DEXL Lab, National Laboratory for Scientific Computing, Brazil
[2] Federal Center for Technological Education of Rio de Janeiro, Brazil
[3] Free University of Bozen-Bolzano, Italy
[4] University of Twente, The Netherlands

Abstract. Digital twin connects concrete systems to digital representations, encoding the real world using software systems, tools and models. Therefore, digital twins should comprise abstractions, formal namings and definitions of categories, properties and relations between concepts, data and entities substantiating one, many or every element of some domain of interest. Considering the possible synergies between digital twins and ontology, and the growing demand for connecting the physical and the virtual world through explicit ontological grounding, our work proposes preliminary discussions about requirements to build an ontology of digital twins. We outline some relevant topics both in the field of digital twins and ontology that are important for the proposal of core reference ontology in the field. We also explore these requirements in detail, from the conception and creation of the virtual environment and the digital twins, to the synchronization between digital world and real world, in addition to computational services, including visualization, prediction, and prescription . Finally, we present topics for future work. [*]

Keywords: Digital twin · Ontology · UFO · Requirements

1 Introduction

Digital twins connect real systems to virtual representations, codifying the physical world using computer systems, tools and models, bringing insight about the current state of a real entity, providing tools to analyse and monitor physical entities from different perspectives, and predicting future states by combining machine learning models and simulations based on first principles. These digital models create the information necessary for decision-making in the real world, assisting applications in the industry, healthcare domain, agriculture and energy [17].

Ontology is the branch of metaphysics that studies different modes of existence, thus, providing theories comprising systems of categories, properties and relations. Foundational ontologies define domain-independent categories about

---

the real world, constituting toolboxes of reusable information modeling primitives for building application ontologies in specific domains [19].

The Unified Foundational Ontology (UFO) is a well-established foundational ontology composed of three main parts: (i) **UFO-A** [11], an ontology of endurants, describing entities including particulars, universals, substances, moments, quality, relations, and situations; (ii) **UFO-B** [15], an ontology of perdurants, defining concepts such as events and time; and (iii) **UFO-C** [13], an ontology of social entities, built on top of UFO-A and UFO-B, to systematize social concepts, such as plan, action, goal, agent, intentionality, and commitment, among others. Grounded in the UFO, several ontologies were proposed to bring notions towards specific domains of applications, such as an ontology of artifacts [23] and ontologies in the context of software development, execution and process [13, 3].

Therefore, it is possible to map systems to their digital twin counterparts through explicit ontological grounding, capturing: (i) fundamental aspects of their existence, such as conception, creation, prototyping, functionalities and disposal; (ii) computational aspects, including development and deployment, data processing, and model execution; and (iii) application-oriented aspects, considering the impact of digital twins in the market, in the business, and in the lifecycle of products and processes. The paper proposes preliminary discussions on requirements to build an ontology of digital twins, mainly focused on foundational aspects, in addition to presenting computational aspects at a higher level.

The paper is organized as follows: Section 2 outlines the fundamentals of digital twins and ontology, presenting topics which are relevant to further discuss the requirements for an ontology of digital twins. Section 3 explores in details these requirements, from the conception of the virtual environment and the digital twins, and their creation, to the synchronization between digital twin and real world, and computational services, including visualization, prediction, and prescription . Section 4 offers an insight into related work. Finally, Section 5 concludes and points to further future work.

2 Fundamentals of Digital Twins

A **digital twin** is a virtual representation of the real world, whose data flow between the physical object and its digital counterpart is integrated and automated in both directions. Therefore, it reflects the conditions of the physical object through data captured from the physical world, and provides feedback by offering computational services including visualization, control, prediction and what-if scenario simulation in the virtual world, existing along the life-cycle of the physical entity [24].

The use of twins to monitor physical objects and test different conditions on them dates back to NASA's Apollo program, simulating the conditions of a vehicle in the space using another vehicle on Earth, mirroring the flight conditions as accurately as possible [24]. In the 1990s, Gelernter [7] described the idea of recreating real world inside a virtual environment, and Grieves first proposed a digital twin as a model in the early 2000s as a concept for Product Lifecycle Management (PLM), asserting that all real-world systems are dual in nature, having both a physical embodiment and a virtual representation [9].

2.1 Digital Twin Models and Platforms

Earlier digital twin models in the literature followed Grieves proposal [8], being composed of three dimensions: (i) a **physical environment**, containing physical entities in real space; (ii) a **virtual environment**, consisting of virtual objects and computational tools in digital space; and (iii) the **data and information connection**, linking the physical and the virtual environments. Furthermore, Grieves and Vickers [10] classifies digital twins into three sub-types: (i) a **digital twin prototype**, which includes requirements and models related to the concept of a physical entity; (ii) a **digital twin instance**, describing a specific physical entity that the instance remains linked to throughout the life-cycle of that entity; and (iii) the **digital twin environment**, an integrated, multi-domain physics application space for operating on digital twins, including the laws of physics and rules that every twin instance in this environment must respect.

A more sophisticated model proposes five dimensions for digital twins, including [17]: (i) a **physical environment**; (ii) a **virtual environment**, regarded as the sum of all models representing physical entities; (iii) **services** to be executed by the Digital Twins, such as model execution, visualization, machine learning prediction, prescriptive task allocation and maintenance; (iv) the **data** to which the Digital Twin has access, coming both from the physical and the virtual environments; and (v) **connections** between every other dimension. Unlike the three-dimensional model, the 5D model separates data and communications, and further includes services within digital twin domain, in general being offered externally to the virtual environment where the digital twin settings and properties are defined, such as in a cloud platform.

2.2 Digital Twin Platforms

The development of a digital twin is a computational workflow composed of computational services representing models for the process stage, and its interactions. Therefore, digital twin platforms should provide an orchestration of these different independent services to provide both flexibility and computational performance. A digital twin platform should provide different levels of abstraction, including [2]: (i) the **digital twin user** level, where the user gains an access to the available digital twins in the form of applications; (ii) the **digital twin developer**, where the platform provides resources for the development of digital twins; (iii) the **computational service developer** level, with the platform providing an API for the development of computational services; and (iv) the **infrastructure provider** level, where instances of the computing services are mapped to computing resources.

2.3 Digital Twin Maturity Levels

A digital twin is the product of an orchestration of computational services and available data connected to their physical counterparts in real-time throughout their life-cycle, and therefore it can be used to monitor the current state of the objects, predict future states, prescribe desired states, and remotely correct real-object states in order to fulfill some real-world requirement. Verdouw et al. [22] proposes five different maturity levels of a Digital Twin according to its functionalities: (i) an **imaginary** twin is a conceptual entity capable of describing an object that does not yet exist, containing information necessary to materialize it, i.e. a digital twin prototype; (ii) a **monitoring** twin is a digital representation of the real state, behaviour and trajectory of a physical object present in the system, similar to a digital twin instance; (iii) a **predictive** twin a digital projection of the future states and behaviours of the physical object using predictive analytics such as statistics, simulation and machine learning methods; (iv) a **prescriptive** twin is an intelligent digital object capable of recommending corrective and predictive actions on the physical object based on optimization algorithms and specialized heuristics; and (v) an **autonomous** twin operates autonomously and fully controls the behaviour of real objects without human intervention, becoming self-adaptive systems able to learn about the environment, perform self-diagnosis and adapt to user preferences.

3 Requirements for an Ontology of Digital Twins

We identified several Competency Questions that ontologies for digital twins should address. For each competency question a possible answer is advanced and brought up for discussion (see Table 1).

3.1 Models and Multi-Level Theory Requirement

Since the virtual environment of a digital twin is composed of models of the physical environment, the first step in developing an ontology of digital twins is to elaborate the concept of a **model** in detail. Following the definition of a model as an **artifact** that abstracts a system or a process from a particular perspective [3], one could notice the distinction between: (i) the conception of modeling, which is regarded as a **model universal**; (ii) a **model** as a representation encompassing information, rules, methods and premises, such as a 3D model or a class model; (iii) a **model of a type** (a **prototype)**, including a 3D model of a car; and (iv) a **model of an individual**, i.e. a 3D model of John's car.

Note that a model universal is a type whose individuals are models, as every model instantiates a particular concept of modeling (e.g., 3D Model, 2D Model). General model universals can be specialized in particular subtypes of model universals that instantiate particular prototypes, (i.e. specifications of properties, dispositions and modes) that a model of an individual must conform to. The same entity in reality can be represented by models instantiating different prototypes (i.e., specific subtypes of model universals). Therefore, a **multi-level**

Table 1. Competency questions for an ontology of digital twins.

#	Question	Sec.
Q1	What is the difference between a model universal, a prototype, and a model of an individual?	3.1
Q2	What is a model of a substantial?	3.1
Q3	How does a model relate to real world entities?	3.1, 3.2
Q4	How to model an event in the real world bringing about a situation in which a substantial in the real world is present?	3.1
Q5	How to represent a moment inhering in a substantial in the real world in a model of that substantial?	3.1
Q6	What is a twin instance?	3.2
Q7	Given a substantial having parts and otherwise related individuals, how does a twin instance of the former relate to twin instances of the latter?	3.2
Q8	Who decides which substantials in the real world are going to be modeled by twins and how?	3.2
Q9	What is a twin prototype?	3.2
Q10	What is the difference between hardware twins and digital twins?	3.3
Q11	How are hardware twin instances or digital twin instances created?	3.3
Q12	How does an execution of a digital twin instance give information about situations in the real world where modeled substantials are present?	3.3, 3.5
Q13	What is a twinning disposition of a digital twin instance?	3.4
Q14	What is a twinning process and how does it manifests a twinning disposition?	3.4
Q15	What is a feedback brought by a digital twin instance and how is it triggered?	3.5
Q16	What are the temporal references in a digital twin context?	3.6
Q17	What is a possible situation?	3.6
Q18	What is a digital sibling instance?	3.6
Q19	How does a digital sibling instance bring feedback about a possible situation?	3.6

theory [5] is a fundamental requirement to delineate the description of a model within an ontology of digital twins, embracing topics including powertypes and categorization schemes which are themselves part of subject matter [12].

Following Guizzardi et al. [12], a prototype may be regarded as a powertype whose instances are variable embodiments which classify instances of models of an individual. A variable embodiment is an individual f that, at each world w, picks up a particular rigid embodiment - a set of individuals standing on a relation - according to a given principle F. As such, a prototype can hold general properties that characterize that variable embodiment including: resultant properties from instances of models of an individual classified by it; regularity properies that capture regularities over the instances of a particular type; and direct properties of the type, but not of any individual instance [12] (see Figure 1).

From now on, we focus on models and prototypes of substantials, i.e. existentially independent individuals. Even though it is possible to model events and

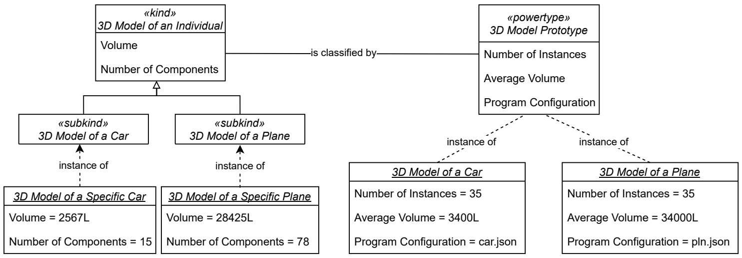

Fig. 1. A UML class diagram showing the relationship between instances of a prototype and instances of models of an individual.

moments, digital twins are usually related to objects (car, spacecraft or an oil platform) or agents (person, company or a society).

An **event** in the real world would bring about a situation in which the substantial of interest participates. Such an event must be mapped to a **corresponding event in the digital world** that would bring about a situation involving the model, which reflects the real situation. Similarly, a **moment inhering in a substantial** can be abstracted into a **moment that inheres in the model of that substantial**.

The term **real-world** in our context refers to the portion of reality which refers to the substantial and other individuals of interest. We take here the real-world to be a complex situation composed by situations in which these individuals participate.

3.2 Twin Instances, Mereology, Relationships and Twin Prototypes

The first use of something to monitor real objects and test possible situations was by the use of an object, a hardware twin assumed identical to the one studied, whose goal was to simulate the exact conditions of the object of interest to obtain a representation of reality. Therefore, before specifying whether a twin is digital or non-digital, a more fundamental definition of a *twin*, regardless of its **physical manifestation**, should be conceived.

Henceforth, the difference between a **twin** and a **model** becomes quite subtle. Considering that a twin is intended to be **expandable**, i.e. being able to integrate, add or replace models [18], another essential requirement is that a twin can be understood as a composition of models. More specifically, a **twin instance** is a **complex model of individuals** composed of one or more models of individuals. As an example, a twin instance of an specific car could be composed of a 3D model of that car, a database table containing values of properties including speed, acceleration and direction of that car, and a predictive model which receives, as an input, images captured from a camera attached on the front of the car, and, as an output, detected objects with their positions and categories.

It is noteworthy that a twin instance could contain models of different individuals, such as in our previous example, where the twin instance of a car contains a model of a camera attached to the car. Nevertheless, the twin instance refers to the car, implying that the camera is, according to the twin, a part of it. Moreover, a twin instance may incorporate other twin instances (i.e. a twin of the camera, which is part of the twin of the car), and an conceptual theory of **mereological relations** is required to implement such particularities of a twin [11], in addition to ensuring the twins are **scalable**, providing insights at different scales [18].

With such discussion, it is not mandatory to create twins of every substantial present in a system, since a model is not necessarily a twin. The decision of which substantials are going to be modeled or even projected into a twin is reflected in the **intention** of a **stakeholder** (who is responsible for the twin instance) whose interest is to represent the system these substantials compose. For instance, if the intention of creating the twin of a car is to control speed, acceleration and direction, one would not necessarily instantiate a twin of the driver. However, if the objective includes modeling impacts from a possible car accident, probably that additional twin would be needed.

One question, however, that arises from this example is: is the driver twin part of the car twin? As the aforementioned substantials have no mereological connection, instead have relationships that range from physical contact to ownership and control (i.e., the driver owns and controls the car), such aspects need to be reflected in their twins as well. Consequently, a theory of relations must be encompassed by an ontology of digital twins in order to faithfully abstract interactions, physical and social bonds in the real world [6].

Similarly, a **twin prototype** can be discussed from the view of a **complex prototype**. In the aforementioned example, a twin prototype of a car could contain the 3D prototype of a car, the schema describing the properties mentioned above (speed, acceleration and direction), and the specifications of a predictive model for the camera. A twin instance, therefore, is a particular exemplification of a twin prototype.

Next, it is essential to discuss how to physically manifest these twins, giving models the capacity to bring information to a **stakeholder** about some part of the reality where the real substantial participates, instantiating the modeled specifications, properties, and dispositions into an object that is considered, by some stakeholder, a reliable representation of the substantial of interest. Due to the history of the development of digital twins, it is interesting to distinguish a **hardware twin** from a **digital twin**.

3.3 Physical Manifestation of a Twin: Hardware Twins and Digital Twins

The **creation** of a hardware or a digital twin instance is described as an **activity occurrence**: (i) requiring **resources**, such as every physical resource to build the substantial of the same type and a hardware twin prototype (for a hardware twin), or a machine to execute software and a digital twin prototype in the case of a digital twin, along with their corresponding twin instance, which refers to the substantial whose representation is intended; (ii) adopting **procedures**, such as a **document template** containing the description of the twin that is going to be implemented either physically or virtually, and a **method**, which are the plan description for implementing the model in its physical manifestation. This activity occurrence receives, as an *input*, a twin instance, and *outputs* a hardware twin instance or a digital twin instance, depending on the chosen resources. These steps require elements of a process ontology, which is more precisely defined in the context of software [13], and of significant interest to the digital twins area.

Although it is easier to check that the **hardware twin instance** is an artifact composed of (i) a substantial of the same type of the one to which the twin instance refers and (ii) the twin instance, a **digital twin instance** is more subtle to define. In fact, it may be understood as an artifact composed of a twin instance and a **program**, implementing a **program specification** intending to satisfy some higher-level **system specification**, in this case regarding the representation of the real substantial. The existence of digital objects and their role in ontologies are important for a fuller definition of digital twins [16].

The Software Ontology (SwO) [3] proposes several details about how a program has physical dispositions to be copied and loaded into a machine, and the execution of a loaded copy is an event which physically manifests the aforementioned disposition. Moreover, the execution brings about an **observable state**, a particular **situation** involving the qualities and values of the machine in which the loaded program copy inheres, as well as of entities residing in that machine. Hence, the execution of a loaded copy of a digital twin instance should bring about an observable state where the twin instance (the models) and its moments (properties and dispositions) participate.

In addition, there is an agent who is involved in the activity occurrences and becomes responsible for the physically-manifested twin, being regarded as a twin developer (further specialized into hardware twin developer or a digital twin developer). In fact, one can argue that a twin developer is a *role* played by a stakeholder when he/she becomes responsible for these physically-manifested twins. Therefore, these intentionally created objects require an ontology of artifacts [23] to fully describe their context of development and use.

Focusing the discussion from now on digital twins, it is necessary to understand in more detail how an observable state brought about by an execution of a loaded copy of a digital twin instance refers to a situation in the real world where the real substantial participates. We need to discuss how, after the creation of a digital twin instance, the state of the real-world substantial and the state of its digital counterpart are synchronized, a process known in the literature as **twinning**.

3.4 The Synchronization of the Real-World Substantial and the Digital Twin Instance

A digital twin instance inheres a disposition to update its current state in order to match one or more moments of the twin instance composing it (such as properties, relations with other substantials, modes or dispositions) with the current state of the substantial it refers to. That disposition is manifested by events associated with the data lifecycle in a digital twin. Thus, a more complete view of this synchronization step is to describe it as software processes [13], whose associated activities relate to the different steps from data collection, transmission, storage and integration, to data processing, cleaning, analysis and mining [1], and the result is an updated state of the digital twin instance.

Twinning may be also interpreted as a commitment that the digital twin has to reliably represent the real substantial. However, as digital twins are typically non-agentive objects, they cannot bear intentional moments (including commitments). They are programmed by a user, which, in turn bears these intentional moments. Instead, we take twinning as a complex disposition reflected in the digital twin disposition to reproduce the state of the real substantial, and the disposition of the real substantial to provide data to help the digital twin update its own state. The different computational steps to perform such synchronization can be expressed through **twinning processes**.

After defining how the real world affects the digital world, the next step is to define different computational services to which the digital twin must have access, and more specifically how the results obtained in the digital world have an impact in the real world. In fact, the state of the digital twin reflects the state of a substantial, so it is essential to describe in detail how such a reflection is perceived by the stakeholder responsible for the digital twin, and what kind of information is inferred from that computational state.

3.5 Feedback about the Real World

A digital twin instance is composed by a program whose execution of its loaded copy brings about an observable state in the computational environment which refers to a situation in the physical environment where the substantial referred by the twin instance participates. Although theoretically this observable state gives a complete information about the real world, in practice, we have: (i) uncertainties of the twinning process that do not guarantee that the observable state of the digital twin instance is mapped identically to the situation in which the substantial participates; and (ii) situations that satisfy the same propositional content of interest as the observable state. For instance, if the execution of the digital twin instance of a specific car brings about the following state: ‘the car is moving with a speed of 50km/h’, ideally there is a situation ‘the real car is moving with a speed of 50km/h’. Nevertheless, other situations could be: (i) the car is moving with a speed between 48km/h and 52km/h (uncertainty), and (ii) the car is below the speed limit allowed by the urban road where it is travelling.

Therefore, one may define the **feedback** brought by a digital twin instance as an **event** that creates a **belief** in the **digital twin user**. A digital twin user is a *role* played by a stakeholder who is interested in a situation where a particular substantial is present (i.e., the observable state given by the execution of the digital twin refers to a situation in which the substantial that twin models is present). As a belief, it can be justified if the situation referred by that observable state obtains.

To trigger a feedback, a digital twin user must run or schedule the execution of programs within a software product that participate in software processes as software resources, resulting in the updated status of the digital twin instance and the feedback event. This requires elements of the Software Ontology (SwO) [3], and Software Process Ontology (which includes activities, resources and scheduled actions [13]).

3.6 Computational Services, Temporal Structure and Possible Situations

To detail the role of **computational services** in a digital twin domain, it is necessary to grasp the importance of **time**, which is addressed in details in UFO-B [15]. There are three temporal references in a problem involving digital twins: (i) the **user temporal reference**; (ii) the **temporal reference of the digital twin instance**, built from the data captured and stored in the computer system; and (iii) the **temporal reference of the modeled substantial**, in terms of the user interest. More specifically, the temporal reference is associated with the time point or the time interval both of the observable state of the digital twin and of the situation in which a substantial is present and about which a stakeholder intends to obtain feedback.

Regarding the observable state of the Digital Twin instance, it is possible both to refer to data obtained in real time (in this case, the time reference of the digital twin is similar to that of the user), as well as to stored historical data (or that is, the digital twin represents an observable state with data obtained in the past) or to simulations (which can either represent a possible observable state in the past or in the future, or even in counterfactual situations). As for the situation of the real entity, it is possible to obtain feedback that leads to a belief that an observable state infers a situation in real time, infers a situation that happened in the past or even a situation that may happen in the future. Several tasks, including **real-time monitoring**, **historical data analytics**, **real-time analytics**, **simulation**, **prediction**, and **prescription**, may be executed as computational services within the virtual environment. Moreover, when the goal behind a task is to get feedback from a situation in the real world that it is not known whether it happened or not, or that might happen in the future, we can consider the existence of **possible situations**. These concepts may be further explored with the help of ontologies addressing simulations aspects [14].

Virtual representations of reality that intend to give feedback about possible (but non-actual) situations cannot, by definition, be called digital twins, despite such a term being used even in this context: These are typically named a **sibling**, a spin-off of the reality encompassing a *possible represented world*. A **digital sibling instance** may be regarded, ontologically, as a special kind of digital artifact, hence an instance of a digital artifact prototype, which is created by using a digital twin instance as an input.

3.7 A Preliminary Conceptual Model of Some Digital Twins Notions

We here propose a very preliminary conceptual model of some Digital Twin notions, connecting these concepts to those of UFO and some of its extensions. This model is show in Figure 2.

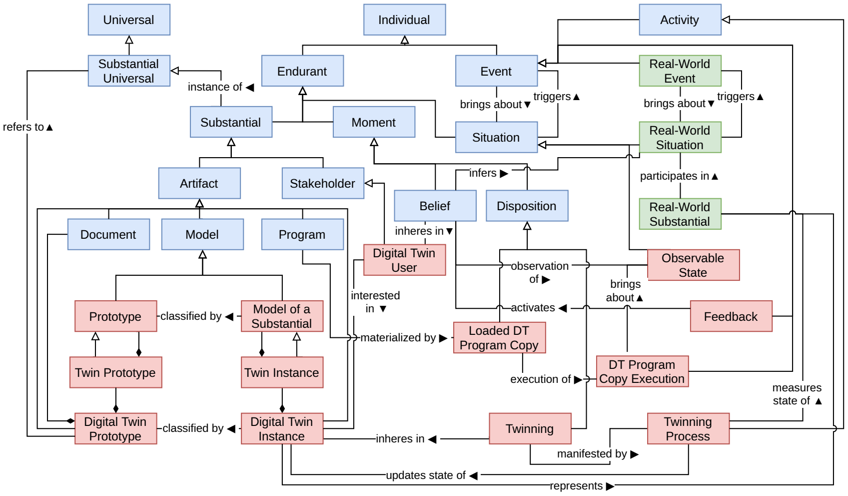

Fig. 2. A digital twin conceptual model encompassing requirements and discussions aforementioned where (i) blue notions are from UFO and ontologies grounded in UFO; (ii) red ones are specific to digital twins, and (iii) green ones refer to real-world elements.

4 Related Work

To the best of our knowledge, no attempts have been made so far to discuss requirements to develop an ontology of digital twins focusing on fundamental aspects of their existence and use. Nevertheless, some authors propose ontologies and conceptual models aimed at applications or more specific computational aspects, such as managing digital twin databases. Barth et al. [1] structures digital twin applications by developing a domain-specific ontology summarized by their data resources, external value creation and internal value creation. Singh et al. [20] capture and model the conceptual knowledge of the digital twin domain by an ontology model further transformed into a mininum data model structure to map, query and manage databases for digital twin applications. Steinmetz et al. [21] propose an ontology for digital twins of devices in an IoT system, providing them with application programming interfaces (APIs) and human-machine interfaces.

5 Conclusion and Future Directions

This work discusses several requirements to build an ontology of digital twins, encompassing different steps from models to the conception of the digital twin, creation of the prototype and materialization of its instances, in addition to some of its main functionalities, such as real-world synchronization, visualization, data analysis, prediction, and simulation. Possible future directions based on this work include:

- Design and implement a core reference ontology of digital twins, using ontologically well-founded languages such as OntoUML [11], and following the Systematic Approach for Building Ontologies (SABiO) [4].
- Further elaborate the relation between ontology, which describes reality, and phenomenology, which seeks to understand causes and effects, in addition to simulate possible worlds using laws of nature modelled by mathematical and computational tools.
- From agent-based ontologies, extend the discussion to include autonomous digital twins.

We expect that our paper motivates the development of novel ontologies of digital twins, specially those grounded in foundational ontologies.

Acknowledgements. This research is partially supported by Accenture Israel Cyber R&D Lab (RiskGraph Project).

References

1. Barth, L., Ehrat, M., Fuchs, R., Haarmann, J.: Systematization of digital twins: Ontology and conceptual framework. In: Proceedings of the 2020 The 3rd International Conference on Information Science and System. pp. 13–23 (2020)
2. Borodulin, K., Radchenko, G., Shestakov, A., Sokolinsky, L., Tchernykh, A., Prodan, R.: Towards digital twins cloud platform: Microservices and computational workflows to rule a smart factory. In: Proceedings of the10th International Conference on Utility and Cloud Computing. pp. 209–210 (2017)
3. Duarte, B.B., de Castro Leal, A.L., de Almeida Falbo, R., Guizzardi, G., Guizzardi, R.S., Souza, V.E.S.: Ontological foundations for software requirements with a focus on requirements at runtime. Applied Ontology **13**(2), 73–105 (2018)
4. Falbo, R.A.: Sabio: Systematic approach for building ontologies. In: Proc. of the Proceedings of the 1st Joint Workshop ONTO.COM/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering. CEUR (2014)
5. Fonseca, C.M., Almeida, J.P.A., Guizzardi, G., Carvalho, V.A.: Multi-level conceptual modeling: Theory, language and application. Data & Knowledge Engineering **134**, 101894 (2021)
6. Fonseca, C.M. et al.: Relations in ontology-driven conceptual modeling. In: International Conference on Conceptual Modeling. Springer (2019)
7. Gelernter, D.: Mirror Worlds: Or the day software puts the Universe in a shoebox... How it will happen and what it will mean. Oxford University Press (1993)
8. Grieves, M.: Digital twin: Manufacturing excellence through virtual factory replication. White Paper **1**, 1–7 (2014)

9. Grieves, M.: Virtually intelligent product systems: Digital and physical twins. In: Flumerfelt, S., Schwartz, K.G., Mavris, D., Briceno, S. (eds.) Complex Systems Engineering: Theory and Practice. American Institute of Aeronautics and Astronautics (2019)
10. Grieves, M., Vickers, J.: Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In: Transdisciplinary Perspectives on Complex Systems, pp. 85–113. Springer (2017)
11. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Ph.D. thesis, University of Twente, The Netherlands (2005), published as the book "Ontological Foundations for Structural Conceptual Models", Telematica Institute Fundamental Research Series No. 15, 05–74, ISBN 90-75176-81-3 ISSN 1388-1795; No. 015; CTIT Ph.D. Thesis, ISSN 1381-3617
12. Guizzardi, G., Almeida, J.P.A., Guarino, N., de Carvalho, V.A.: Towards an ontological analysis of powertypes. In: JOWO@ IJCAI (2015)
13. Guizzardi, G., de Almeida Falbo, R., Guizzardi, R.S.: Grounding software domain ontologies in the unified foundational ontology (ufo): The case of the ode software process ontology. In: CIbSE. pp. 127–140. Citeseer (2008)
14. Guizzardi, G., Wagner, G.: Dispositions and causal laws as the ontological foundation of transition rules in simulation models. In: 2013 Winter Simulations Conference (WSC). pp. 1335–1346. IEEE (2013)
15. Guizzardi, G. et al.: Towards ontological foundations for the conceptual modeling of events. In: International Conference on Conceptual Modeling. pp. 327–341. Springer (2013)
16. Hui, Y.: On the existence of digital objects, vol. 48. U of Minnesota Press (2016)
17. Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L., Nee, A.Y.: Enabling Technologies and Tools for Digital Twin. Journal of Manufacturing Systems (2019)
18. Schleich, B., Anwer, N., Mathieu, L., Wartzack, S.: Shaping the digital twin for design and production engineering. CIRP Annals - Manufacturing Technology **66**(1), 141–144 (2017). https://doi.org/10.1016/j.cirp.2017.04.040
19. Schneider, L.: How to build a foundational ontology. In: Annual Conference on Artificial Intelligence. pp. 120–134. Springer (2003)
20. Singh, S., Shehab, E., Higgins, N., Fowler, K., Reynolds, D., Erkoyuncu, J.A., Gadd, P.: Data management for developing digital twin ontology model. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture p. 0954405420978117 (2020)
21. Steinmetz, C., Rettberg, A., Ribeiro, F.G.C., Schroeder, G., Pereira, C.E.: Internet of things ontology for digital twin in cyber physical systems. In: 2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC). pp. 154–159. IEEE (2018)
22. Verdouw, C., Tekinerdogan, B., Beulens, A., Wolfert, S.: Digital twins in smart farming. Agricultural Systems **189**, 103046 (2021)
23. Weigand, H., Johannesson, P., Andersson, B.: An artifact ontology for design science research. Data & Knowledge Engineering **133**, 101878 (2021)
24. Zhang, H., Zhang, G., Yan, Q.: Digital twin-driven cyber-physical production system towards smart shop-floor. Journal of Ambient Intelligence and Humanized Computing **10**(11), 4439–4453 (2019)