CEUR-WS.org/Vol-3507/paper2.pdf

Emerging Challenges in Compositionality and Correctness for Digital Twins

Eduard Kamburjan, Vidar Norstein Klungre, S. Lizeth Tapia Tarifa, Rudolf Schlatte, Martin Giese, Einar Broch Johnsen and David Cameron

Department of Informatics, University of Oslo, Oslo, Norway

Abstract

A digital twin is an information system that analyzes the behavior of a physical or digital system by connecting streams of observations to dynamic (e.g., simulation) and static (e.g., asset management) models of this twinned system. In large-scale industrial settings, the digital twin will often need to manage a multitude of models for subsystems reflecting different engineering disciplines, vendors, etc. To analyze such complex systems, digital twins must ensure the correct composition of these models and their correct exposure to the user. For the integration and transfer of information between models, digital twins may profit from a formalization of domain knowledge using ontologies, which have proven effective to unify data models. However, it is an open challenge to formalize and verify the correctness of digital twins. This paper discusses this problem for digital twins and illustrates challenges for formal methods with a focus on the composition of heterogeneous dynamic models.

Keywords

Semantic Technologies, Asset Models, Digital Twins

1. Introduction

Digital twins, originally conceived for NASA's space programme [1], enable industry to significantly improve the *life-cycle management* of physical assets. The vision of digital twins is to create a digital replica (the "digital twin"), which is connected in real time to the modelled, traditionally cyber-physical, system (the "twinned system"). Via this real-time connection, the digital twin aims to provide insights into the twinned system's state or behavior.

At the core of this vision, the digital twin coordinates data exchange between (a) the twinned system, (b) a range of model-based analysis tools and (c) stake-holders like engineers and analysts. The data about the twinned system typically combine static asset models and time-series measurements (e.g., data streams from sensors). The analysis tools typically combine simulators of physical models with executable software models. The digital twin computes an approximation of the behavior of the twinned system to explore "what-happened", "what-may-

FMDT'23: Workshop on Applications of Formal Methods and Digital Twins, March 13, 2023, Lübeck, Germany
eduard@ifi.uio.no (Eduard Kamburjan); vidarkl@ifi.uio.no (Vidar Norstein Klungre); sltarifa@ifi.uio.no (S. Lizeth Tapia Tarifa); rudi@ifi.uio.no (Rudolf Schlatte); martingi@ifi.uio.no (Martin Giese); einarj@ifi.uio.no (Einar Broch Johnsen); davidbc@ifi.uio.no (David Cameron)
0000-0002-0996-2543c (Eduard Kamburjan); 0000-0003-1925-5911 (Vidar Norstein Klungre); 0000-0001-9948-2748 (S. Lizeth Tapia Tarifa); 0000-0001-5601-5517 (Rudolf Schlatte); 0000-0002-2058-2728 (Martin Giese); 0000-0002-1172-478X (Einar Broch Johnsen); 0000-0001-5382-3949 (David Cameron)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

happen" and "what-if" scenarios. The engineer can interact with the digital twin to access data, but also to perform more involved operations, e.g., to predict the consequences of changing system parameters, replacing components in the twinned system, or evaluate newly developed designs [2].

A digital twin is a composed, data-intensive system that needs to coordinate its analysis tools, data exchange between the twinned system and models that are relevant for a particular analysis, as well as between different models if necessary. If the twinned system is physical, it consists of a *cyber-physical system* (CPS) in a *physical environment*, i.e., physical boundary conditions (e.g., temperature or fluid pressure) and modelled external actions (e.g., motion tracking devices). In the digital twin, both the CPS and the environment may be modeled by several components, each reflecting a part of the CPS or the dynamics of the operational environment. These smaller, targeted models are typically created by domain experts (e.g., chemical, mechanical or electrical engineers). Digital twins in industry are built from proprietary black-box applications, supplied by the vendor of the component. This limits the possibility to automate workflows within digital twins and to use formal tools to ensure basic correctness properties.

Nonetheless, digital twins are suited for formalization because of the inherent connection to model-based concepts. Challenges arise, besides black-box simulation, from the connection of complex data with complex dynamic models within the digital twin. Observe that there is a dichotomy between correctness for static and dynamic models: the integration of diverging static models can be achieved using semantic technologies, while the correct behavior and compositional constraints for dynamic models can be ensured using formal methods. A crucial step towards the formalization of digital twins is to connect these two approaches and formalize data propagation inside the twin.

While digital twins are often discussed from a data or business perspective [3], we take the formalization perspective in this paper to discuss the connection between static models of data and the composition of dynamic models. Correct data propagation between (and within) diverse models is related to orchestration in co-simulation [4], which is usually restricted to static structure. To configure a co-simulation system correctly for a particular analysis, different simulators need to be orchestrated to exchange data correctly. One particular approach to solve the challenge outlined above, is to combine knowledge graphs with orchestration in generalized co-simulation to ensure correctness. In this article we illustrate this, and further emerging challenges for formal methods with respect to integrating asset models and semantic technologies for digital twins.

Related work. Semantically lifted programs integrate static models represented using semantic technologies and dynamic models such as simulation units, into a programming language [5, 6]. They have been applied to digital twins [7], but correctness has only been considered for specific applications [8, 7]. Recent co-simulation surveys identify a lack of research into modular, stable, and accurate coupling of simulators in dynamic scenarios [4, 9]. There is a long tradition to use semantic technologies to integrate data [10], in the digital twin context this is recently discussed [2, 11, 12].

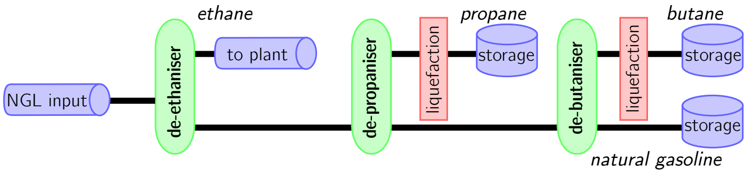

Figure 1: Natural gas liquid fractionation plant, from [17].

2. Background

We briefly review the main concepts in co-simulation and ontologies, which form the basis for our discussion of digital twins and semantic technologies.

Co-simulation denotes a way to implement global simulation of a complex system via the composition of various dynamic models representing the system's components [4]. Each individual model, or *simulation unit*, can be seen as a black box capable of exhibiting behavior, consuming inputs and producing outputs. Assembling these simulation unit into a co-simulation poses some specific coordination challenges. The models must be synchronized not only wrt. the values they exchange (typically via point-to-point typed channels), but also on the current simulation time and when and by how much to advance time.

The time model of simulations, and hence co-simulations, can be *discrete* or *continuous*. In discrete event simulations, a simulation unit synchronizes with the environment at specific timestamps to exchange values. If two events happen at the same time, both are processed before the simulated time progresses. In continuous time simulation (e.g., for physical state), the state evolves continuously, which introduces flexibility in the step size of the time synchronization. For co-simulation scenarios which combine discrete and continuous parts, the orchestrator needs to reconcile the different assumptions about the inputs and outputs of each unit to retain the properties of the constituent systems.

Semantic Technologies [13] are techniques to formally attach meaning to data which can be used when constructing complex intelligent systems such as digital twins [12]. These techniques are based on *ontologies*: formal, conceptual descriptions of a domain, usually expressed in the *Web Ontology Language* (OWL) [14]. The ontology specifies the vocabulary of *classes* and *properties* that can be used by the system model, and a set of *axioms*, i.e., constraints, to which the model must adhere. Ontologies are used in many different domains, both within organizations, and as parts of large open projects, like SNOMED CT, an open ontology for clinical terms [15]. By introducing instances and combining them with classes and properties from the ontology, one can construct statements using the *resource description framework* (RDF) [16]; e.g., to model a concrete storage tank in some facility, one can assign an identifier (`:st1`) to the storage tank instance and connect it to the storage tank class (`:StorageTank`) given in the ontology: `:st1 a :StorageTank.` There is good tool support to check consistency of the resulting knowledge graphs (e.g., *do all axioms indeed hold?*), query them (e.g., *what are all the storage tanks?*), and reason over them to infer new facts, or check if concrete facts are implied.

3. A Simple Engineering Model

We use a small example to illustrate the challenges for coordinating models inside a digital twin. Fig. 1 shows the structure of a natural gas liquid (NGL) fractionation plant. Its input is natural gas liquids, or condensate, which is a mixture of light hydrocarbons (ethane, propane and butane). The purpose of the plant is to separate the light hydrocarbons. In the plant, the natural gas liquids are fed into distillation columns to isolate a single product: ethane, propane, then butane. Each column outputs two streams: a top product gas and the bottoms product that contains the remaining heavier hydrocarbons. The light gas products are either directly sent to a consumer (ethane is, e.g., used as a feed for petrochemicals plants or is burnt as fuel), or they are liquefied for sale.

Distillation is an expensive and energy-intensive process. Operating the plant requires us to monitor the fractionation process and determine optimal parameters like reflux rates and operating pressures for each distillation column and liquefaction unit. We can use dynamic models for simulation, based on non-linear systems of differential equations [18]. Model composition is constrained by domain knowledge about chemistry, thermodynamics and design practice. The parameters of the distillation and liquefaction units depend on the expected properties of the feed stock and constraints on the quality of the processed products. They are selected at design time to optimize the cost and performance of the plant. These parameters may be continuous variables (diameter of a column) or integers (number of trays in a distillation column).

Ontology. An ontology for the fractionation plant can include two main classes: `:Pipeline` and `:Component`; each `:Component` must be either a `:Separator`, a `:StorageTank`, or a `:Liquefier`. Pipelines and components are connected by pipes, captured in the ontology with the object property `:isConnectedTo`. The part of the ontology concerned with separators and pipelines is then as follows:

```
:Component a owl:Class.     :Separator a owl:Class.
:Pipeline a owl:Class.      :isConnectedTo a owl:ObjectProperty.
```

Using this ontology, we construct the pipeline from the feed source (`:pipeline1`) and its connection to the de-ethaniser (`:separator1`) (see Fig. 1) as follows:

```
:pipeline1 a :Pipeline.      :separator1 a :Separator.
:pipeline1 :isConnectedTo    :separator1.
```

The model described here can be part of a digital twin, which additionally ensures correct data exchange and consistency both within the model *and* in its relation to the physical asset. (Remark that our terminology of a digital twin is sometimes called a digital twin architecture or a digital twin environment [19].) In particular, the digital twin must ensure correct data exchange not only between a dynamic model and a twinned system, but between different possible compositions of dynamic models, each running a different "what-if" scenario. These composed models cannot be used to control the twinned system, yet are connected to data streams from it, and possibly to the controlling model — it is paramount to keep explorative models connected to the controlling model, such that these do not influence the behavior of the twinned system directly.

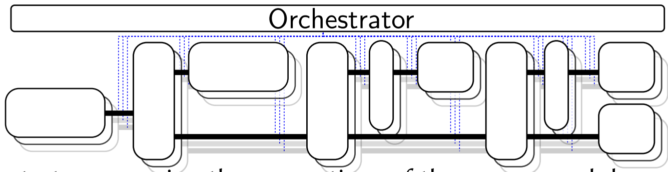

Figure 2: An orchestrator managing the connections of three composed dynamic models, based on Fig. 1; each box is a black-box simulator or a connection to a twinned asset.

4. Challenges

Development and formalization of digital twins beyond the industrial state of the art poses a number of challenges to the technologies employed. We identify two core challenges: Formalizing (a) the *correctness of digital twins*, and (b) the *principles of model composition* for a targeted physical or digital system.

Consider Fig. 2, which shows the structure of models that make up the digital twin. Note that the orchestrator need not be monolithic and that each of the boxes may be a dynamic model or a data stream from a twinned system. At each point in time, several dynamic model compositions may exist, with different configurations, for different purposes. Challenge (a) is to ensure that these composed dynamic models are internally consistent (i.e., they form meaningful co-simulation models), consistent with respect to the domain (i.e., they form models of some possible asset) and consistent with the actual twinned asset (i.e., the composed models and the twinned asset have the same structure).

The Semantics of Composition. Beyond checking for datatype violations and unconnected ports, the modeler must assess whether the composition of dynamic models is meaningful. Further checks are necessary, for example, whether the dynamic models are connected correctly with respect to the existing domain. In Fig. 2, the two output ports of each fractionation unit will have the same data type and physical unit (e.g., pressure or flow) but different semantic meanings; such consistency is a correctness property that relies on domain knowledge.

Static and Dynamic Topologies. The above challenge generalizes beyond connections: If the co-simulation is mirroring an asset (or asset model), then every meaningful component of the asset should be included in the co-simulation. Ensuring that the topology of the configuration is consistent with the domain must, again, take domain knowledge into account.

Observe that the notion of a digital replica touches on coordination aspects of self-organization [20], which must ensure that changing structure adheres to its domain constraints: The structure of the twinned system may change, e.g., due to planned maintenance (some components are shut off and exchanged) or unplanned repair. Tracking such changes is typically not supported by co-simulation frameworks or existing industrial practice such as [21], yet structural reconfiguration is crucial in the digital twin to be able to use historical data without restarting the simulation system.

In our example, this corresponds to three scenarios: (a) Is the dynamic model indeed a replica of an existing system? (b) For a what-if analysis: is the modeled system a possible fractionation plant? (c) For a maintenance analysis: does the proposed modification, adhere to the domain model? E.g., in the NGL example, if more information about `:separator1` and the connected tanks is available, we can use the representation of domain knowledge in an ontology to deduce whether the system adheres to the domain model.

Coordinating Speculative Analyses. The last challenges are concerned with one dynamic model, but as the digital twin moves from reproducing "what-happened" scenarios, in which the factual observations of the twinned system are known, to exploring possible "what-if" scenarios for its future behavior, the knowledge supplied by the twinned system decreases. E.g., one may want to explore how replacing a distillation column, or high environmental temperature, would affect the production of the plant as a whole. In these scenarios, there may be many solutions to the composition problem and the digital twin may need to speculatively explore and *coordinate* different possible solutions. Several of the composed models depicted in Fig. 2 may coexist; the composed models must share the connection to the twinned system and may also share some models.

Asset Models. The structural correctness of a dynamic model with respect to a twinned system requires that the twinned system already has a formal representation. One approach is to use *asset models* and *semantically lifted programs* to uniformly represent the twinned system and the dynamically composed model [7]. In particular for twinned physical systems, asset models can play a central role to achieve correctness and compositionality for digital twins: they formally describe requirements and topologies from the asset's perspective, thereby providing the twin with static configuration data for model composition [22].

An asset model is an organized description of the composition and properties of an asset [23, 24, 25], used to support, e.g., maintenance operations on an asset. Asset models may be formalized as ontologies [26] or use them [27, 28, 29], with semantic data access being a current research focus [30, 31, 26]. We are in particular interested in *top-down asset models* which start by modeling the desired functionality of a system as a whole, and then decompose the system into functional sub-systems. This approach, which relates to model-driven engineering [32], is supported by modelling tools and languages such as SysML (e.g., [33]). A top-down model provides a scalable framework for tracking requirements along a system decomposition and linking requirements to individual components to higher-level system requirements [34, 35]. We conjencture that top-down asset models can be used to tackle further challanges, by enriching them with information specific to digital twins.

5. Conclusion

Digital twins connect the management and development of a physical or digital system by applying analyses to a digital model in real time. In large-scale industrial settings, the asset is captured by a multitude of models, which stem from different engineering disciplines, different domain models and different vendors. Digital twins need to correctly integrate and exchange data between such models. This paper discusses challenges for correctness and compositionality in the setting of digital twins, and proposes the use of asset models and formalized domain knowledge to enable formal methods to meet these challenges.

Acknowledgments

This work was supported by the Norwegian Research Council via the SIRIUS Centre (237898) and the PeTWIN project (294600).

References

[1] E. Glaessgen, D. Stargel, The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles, AIAA, 2012. URL: https://doi.org/10.2514/6.2012-1818.

[2] D. Cameron, A. Waaler, T. M. Komulainen, Oil and gas digital twins after twenty years. How can they be made sustainable, maintainable and useful?, in: SIMS 59, 2018. doi:10.3384/ecp181539.

[3] B. R. Barricelli, E. Casiraghi, D. Fogli, A survey on digital twin: Definitions, characteristics, applications, and design implications, IEEE Access 7 (2019). doi:10.1109/ACCESS.2019.2953499.

[4] C. Gomes, C. Thule, D. Broman, P. G. Larsen, H. Vangheluwe, Co-simulation: A survey, ACM Comput. Surv. 51 (2018). URL: https://doi.org/10.1145/3179993. doi:10.1145/3179993.

[5] E. Kamburjan, E. B. Johnsen, *Knowledge Structures over Simulation Units*, in: ANNSIM, IEEE, 2022.

[6] E. Kamburjan, V. N. Klungre, R. Schlatte, E. B. Johnsen, M. Giese, *Programming and Debugging with Semantically Lifted States*, in: ESWC, volume 12731 of *LNCS*, Springer, 2021.

[7] E. Kamburjan, V. N. Klungre, R. Schlatte, S. L. T. Tapia, D. Cameron, E. B. Johnsen, *Digital Twin Reconguration Using Asset Models*, in: ISoLA, volume 13704 of *LNCS*, Springer, 2022.

[8] E. Kamburjan, C. C. Din, R. Schlatte, S. L. T. Tapia, E. B. Johnsen, *Twinning-by-Construction: Ensuring Correctness for Self-Adaptive Digital Twins*, in: ISoLA, volume 13701 of *LNCS*, Springer, 2022.

[9] G. Schweiger, C. Gomes, G. Engel, I. Hafner, J. Schoeggl, A. Posch, T. S. Nouidui, An empirical survey on co-simulation: Promising standards, challenges and research needs, Simul. Model. Pract. Theory 95 (2019). URL: https://doi.org/10.1016/j.simpat.2019.05.001. doi:10.1016/j.simpat.2019.05.001.

[10] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Ontology-based data access and integration, in: L. Liu, M. T. Özsu (Eds.), Encyclopedia of Database Systems, Springer New York, New York, NY, 2018. URL: https://doi.org/10.1007/978-1-4614-8265-9_80667. doi:10.1007/978-1-4614-8265-9_80667.

[11] J. M. Rozanec, L. Jinzhi, A. Kosmerlj, K. Kenda, K. Dimitris, V. Jovanoski, J. Rupnik, M. Karlovcec, B. Fortuna, Towards actionable cognitive digital twins for manufacturing, in: SeDiT@ESWC, volume 2615 of *CEUR*, CEUR-WS, 2020.

[12] E. Kharlamov, F. Martín-Recuerda, B. Perry, D. Cameron, R. Fjellheim, A. Waaler, Towards semantically enhanced digital twins, in: BigData, IEEE, 2018.

[13] P. Hitzler, M. Krötzsch, S. Rudolph, Foundations of Semantic Web Technologies, Chapman and Hall/CRC Press, 2010.

[14] W3C, OWL Working Group, Web ontology language, ???? https://www.w3.org/OWL.

[15] SNOMED International, SNOMED CT, ???? https://www.snomed.org/.

[16] W3C, RDF Working Group, Resource description framework, ???? https://www.w3.org/RDF.

[17] F. Jahn, M. Cook, M. Graham, Hydrocarbon Exploration and Production, Developments in Petroleum Science, 2nd ed., Elsevier, 2008.

[18] A. Rahimi, M. Mustafa, M. Zaine, N. Ibrahim, K. Ibrahim, N. Yusoff, E. Al-Mutairi, M. Abd.Hamid, Energy efficiency improvement in the natural gas liquids fractionation unit, Chemical Engineering Transactions 45 (2015). URL: https://www.cetjournal.it/index.php/cet/article/view/CET1545313. doi:10.3303/CET1545313.

[19] D. Jones, C. Snider, A. Nassehi, J. Yon, B. Hicks, Characterising the digital twin: A systematic literature review, CIRP Journal of Manufacturing Science and Technology 29 (2020). URL: https://www.sciencedirect.com/science/article/pii/S1755581720300110. doi:https://doi.org/10.1016/j.cirpj.2020.02.002.

[20] A. Steiniger, A. M. Uhrmacher, Composing variable structure models - A revision of COMO, in: T. I. Ören, J. Kacprzyk, L. Þ. Leifsson, M. S. Obaidat, S. Koziel (Eds.), SIMULTECH, SciTePress, 2013. URL: https://doi.org/10.5220/0004486302860293. doi:10.5220/0004486302860293.

[21] OSIsoft, The Power of Asset Framework, Technical Report, OSIsoft, LLC, 2015. URL: https://resources.osisoft.com/white-papers/the-power-of-asset-framework/, published at https://resources.osisoft.com/white-papers/the-power-of-asset-framework/.

[22] D. B. Cameron, A. Waaler, T. M. Komulainen, Oil and Gas digital twins after twenty years. How can they be made sustainable, maintainable and useful?, in: SIMS 59, Linköping University Electronic Press, 2018. URL: http://dx.doi.org/10.3384/ecp181539. doi:10.3384/ecp181539.

[23] J. Heaton, A. K. Parlikad, Asset information model to support the adoption of a digital twin: West cambridge case study, IFAC-PapersOnLine 53 (2020). URL: https://www.sciencedirect.com/science/article/pii/S2405896320302093. doi:https://doi.org/10.1016/j.ifacol.2020.11.059.

[24] M. Rotondi, A. Cominelli, C. Di Giorgio, R. Rossi, E. Vignati, B. Carati, The benefits of integrated asset modelling: lessons learned from field cases, in: Europec/EAGE Conf. and Exhibition, OnePetro, 2008.

[25] K. Wei, J. Z. Sun, R. J. Liu, A review of asset administration shell, in: IEEM, 2019. doi:10.1109/IEEM44572.2019.8978536.

[26] E. Fjøsna, A. Waaler, READI Information modelling framework (IMF). Asset Information Modelling Framework, Technical Report, READI Project, 2021. URL: https://readi-jip.org/wp-content/uploads/2021/03/Information-modelling-framework-V1.pdf.

[27] R. Mehmandarov, A. Waaler, D. Cameron, R. Fjellheim, T. B. Pettersen, A semantic approach to identifier management in engineering systems, in: Big Data, IEEE, 2021. doi:10.1109/BigData52589.2021.9671515.

[28] A. Poggi, D. Lembo, D. Calvanese, G. D. Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, J. Data Semant. 10 (2008). doi:10.1007/978-3-540-77688-8_5.

[29] M. G. Skjæveland, M. Giese, D. Hovland, E. H. Lian, A. Waaler, Engineering ontology-based access to real-world data sources, J. Web Semant. 33 (2015). doi:10.1016/j.websem.

2015.03.002.

[30] M. Wiedau, L. von Wedel, H. Temmen, R. Welke, N. Papakonstantinou, ENPRO Data Integration: Extending DEXPI Towards the Asset Lifecycle, Chemie Ingenieur Technik 91 (2019). doi:10.1002/cite.201800112.

[31] IOGP JIP 36, CFIHOS Standards, https://www.jip36-cfihos.org/cfihos-standards/, ???? Accessed: 2021-12-12.

[32] J. Bickford, D. L. Van Bossuyt, P. Beery, A. Pollman, Operationalizing digital twins through model-based systems engineering methods, Systems Engineering 23 (2020). doi:10.1002/sys.21559.

[33] C. Nigischer, S. Bougain, R. Riegler, H. P. Stanek, M. Grafinger, Multi-domain simulation utilizing SysML: state of the art and future perspectives, Procedia CIRP 100 (2021). URL: https://www.sciencedirect.com/science/article/pii/S2212827121005382. doi:10.1016/j.procir.2021.05.073.

[34] P. Delgoshaei, M. A. Austin, D. A. Veronica, A Semantic Platform Infrastructure for Requirements Traceability and System Assessment, in: ICONS, IARIA, 2014. URL: http://rgdoi.net/10.13140/2.1.5090.9446. doi:10.13140/2.1.5090.9446.

[35] A. Fraga, J. Llorens, L. Alonso, J. M. Fuentes, Ontology-Assisted Systems Engineering Process with Focus in the Requirements Engineering Process, in: Complex Systems Design & Management, Springer, 2015. doi:10.1007/978-3-319-11617-4_11.