



# Modeling of, for, and with digital twins

Jeff Gray<sup>1</sup> · Bernhard Rumpe<sup>2</sup>

Published online: 21 September 2022  
© The Author(s) 2022

Have you heard this one? *Two identically looking boys walk into their new class. The teacher looks into the class roster and observes that both students also have the same birthday, mother and father. The teacher says, “You must be twins,” but the boys reply, “No sir, we are not.”*

How can this be? Please read on.

There exists a plethora of definitions for **Digital Twins** (DTs). Although there is a rough convergence toward a common definition, there is no consensus about what a digital twin actually comprises. We have found differences in the definition given in publications, such as issues of narrowness, by focusing on specific use cases, domains, or technologies in the definition. The most prominent and earliest manifestation of the digital twins concept was simulation of the physical twin (or more precisely the Cyber-Physical System, consisting of a physical part and an embedded part for controlling software).

It seems natural to argue that the digital twin accompanies the physical twin during operation, as well as in the early engineering phase, during maintenance and after the system is retired. As a consequence, the RWTH Cluster of Excellence “Internet of Production” suggested the following definition for a digital twin [DMR+20]:

**Definition** A **digital twin** of a system consists of

- A set of **models** of the system,
- A set of **digital shadows**, and
- Provides a set of **services** to use the data and models purposefully with respect to the original system.

To complement the definition, a **digital shadow** includes a set of contextual data-traces with their aggregation and abstraction collected for a specific purpose with respect to the original system. This definition comprises all kinds of collectable data through sensors, user input, and observations of internal state and flowing signals.

To be able to create a digital twin requires that there are observable elements in the physical world that can be monitored, sensed, and possibly also actuated and eventually controlled. The controlling concept is a particularly sensitive aspect because it touches all kinds of safety and security issues. However, we want to highlight another aspect about digital twins in this editorial, namely, the important concept of how typical engineering models and digital twins interact.

From our first observation, the *set of engineering models* helps to understand the system in the physical world, e.g., structure, behavior, physical, geometrical or mathematical models. These models about the physical twin might be very usable, e.g., to create parts of the digital twin. For example, collecting data in digital shadows is one possibility, but if the internal structure or the desired behavior of a physical component is not known, it is more difficult to evaluate the collected and aggregated data than if the collected data are compared to the intended outcome documented in the engineering models. This allows stakeholders to observe and potentially automatically react to illegal states, as well as optimize any suboptimal behavior of a system. DevOps approaches to system development may benefit greatly from such a close connection between physical and digital twins.

Engineering models are also a good way to connect the collected sensor data to their original sources. In this case, parts of the engineering models become meta-data for the digital shadows. Some parts of the engineering models will in the future potentially not describe a fixed part of the overall system, but might suggest possibilities for configuration. In particular, behavior can be adapted when certain forms of models (e.g., activity diagrams, state machines, Petri nets or BPMN models) are dynamically interpreted by the digital twin and thus amenable for adaptation. These kinds of adaptable models contribute to the current discussions on

---

✉ Bernhard Rumpe  
bernhard.rumpe@sosym.org  
Jeff Gray  
jeff.gray@sosym.org

<sup>1</sup> University of Alabama, Tuscaloosa, AL, USA

<sup>2</sup> RWTH Aachen University, Aachen, Germany

LowCode and NoCode approaches, where high-level models replace low-level coding. This is a beneficial situation, because (1) it allows non-experts to interact with the system more intuitively, and (2) the restricted expressiveness of a modeling language allows better consistency checks upfront than pure code approaches.

From our second observation, a digital twin of a complex system may also represent a complex software system on its own. It might be useful and helpful to use software models to design this kind of system. A key open question is, *How and to what extent do the software models of the digital twin relate to the engineering models of the physical system?* Furthermore, for both the physical and digital twin, the development processes radically differ. How do we entangle both in such a way that neither hinders the other one's successful development and use?

For our third observation, when we closely look at the definition of a digital twin, we can see that the digital twin itself shares some characteristics with the definition of a model. In both cases, (1) there is an original, i.e., the physical twin, (2) the digital twin and the model are abstractions, and (3) the digital twin and the model have a purpose with respect to the original, that is, they should have some causal connection.

Pushing this analogy further would mean that a digital twin is a form of a model of the system. So far, with the term "model" we have mainly considered a passive representation like drawings in appropriate UML/SysML/DSL tools. Digital twins represent a possibly new kind of model amounting to the various already existing kinds of models, such as mental models, executable models, machine learning models, and fashion models (which are at least models in Aristotle's sense).

Returning back to the riddle of our opening paragraph... The two boys explain to their teacher, "We are only two of triplets." Thus, when we speak of digital twins, we imply that there can only be one digital twin per physical representation. No more. Even though the digital twin may grow and change behavior over time—quite like the physical twin's evolution. We think this analogy is foundational to the core of the semantics of digital twins.

PS: [DMR+20] can be found in <https://rumpe.github.io/publications> or in *Conceptual Modeling*, G. Dobbie, U. Frank, G. Kappel, S. W. Liddle, H. C. Mayr (Eds.), Springer International Publishing, Oct. 2020.

## 1 Content of this issue

### 1. Erratum

- "Correction to: Low-code development and model-driven engineering: Two sides of the same coin?" by Davide Di Ruscio, Dimitris Kolovos, Juan de Lara,

Alfonso Pierantonio, Massimo Tisi, and Manuel Wimmer

### 2. Expert Voice

- "An executable metamodel refactoring catalog" by Lorenzo Bettini, Davide Di Ruscio, Ludovico Iovino, and Alfonso Pierantonio

### 3. MODELS 2020 Special Section

Guest Editors: Silvia Abrahão, Juan de Lara, Houari Sahraoui, and Eugene Syriani

### 4. Theme Section on Modeling in Low-Code Development Platforms

Guest Editors: Davide Di Ruscio, Esther Guerra, and Massimo Tisi

### 5. Regular Papers

- "A search-based approach for detecting circular dependency bad smell in goal-oriented models" by Mawal A. Mohammed, Mohammad Alshayeb, and Jameleddine Hassine
- "Recommendations for visual feedback about problems within BPMN process models" by Vinicius Stein Dani, Carla Maria Dal Sasso Freitas, and Lucineia Heloisa Thom
- "DSMCompare: domain-specific model differencing for graphical domain-specific languages" by Manouchehr Zadahmad, Eugene Syriani, Omar Alam, Esther Guerra, and Juan de Lara

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.