EDITOR **PHIL LAPLANTE**
The Pennsylvania State University;
plaplante@psu.edu

Trusting Digital Twins

Phil Laplante, The Pennsylvania State University

A digital twin is a virtual representation of real-world entities and processes synchronized at a specified frequency and fidelity. This new technology has the same trust concerns as do other systems, and we explore them here.

Our current virtual world seems little different than it was in past decades when we talked about specifications and software, both of which are virtual. Specifications and software were key software engineering topics back then and still are today. Here, I explore a new technology termed *digital twins*, and ask, "What has really changed?" The trust concerns are essentially the same, so let's revisit them.

For those of you who are unfamiliar with "digital twins," the Digital Twin Consortium, which is currently developing a digital twin standard, has offered the following description for digital twins:

> "A digital twin is a virtual representation of real-world entities and processes synchronized at a specified frequency and fidelity.
> - Digital twin systems transform business by accelerating holistic understanding, optimal decision-making, and effective action.
> - Digital twins use real-time and historical data to represent the past and present and simulate predicted futures.
> - Digital twins are motivated by outcomes, tailored to use cases, powered by integration, built on data, guided by domain knowledge, and implemented in IT/OT systems."[1]

We won't focus on that definition here because there are others. Instead, we will focus on the 14 trust considerations that were proposed in Voas et al. 2021.[2] These 14 should be considered by any proposed definition for digital twin.

These proposed trust concerns are not directly focused on risk assessment and mitigation, but rather on trust. That is, will digital twin technology provide the desired operational functionality with an acceptable level of quality? Answering this question begins with an understanding of trust. Here, trust is the probability that the intended behavior and the actual behavior are equivalent given a

Digital Object Identifier 10.1109/MC.2022.3149448
Date of current version: 4 July 2022

fixed context, fixed environment, and fixed point in time. Trust is viewed as a level of confidence.

Trust should be considered at several levels. 1) Is the digital twin functionally equivalent to the physical object? 2) Can a specific digital twin be composed with another digital twin? 3) Is enough information available about the environment and context of the physical object? 4) Can digital twin technology be standardized to the point where certification of a digital twin is possible? Now, let's explore the 14 trust issues.

DIGITAL TWIN CREATION ORDERING

The point in time at which a digital twin is created will have an impact on the correctness of the digital twin. For example, is it created before the physical object is created, or is it reverse-engineered from the physical entity (that it is intended to mirror)? Both approaches are valid. However, the fidelity of the digital twin may be reduced if it is created after the physical entity exists because there may be internal unknowns about the existing physical entity that cannot be discovered. A good analogy here is commercial off-the-shelf software. Such products are black boxes—the source code is usually unavailable to the customer or integrator and, thus, hides internal syntax. For digital twins, this is a trust consideration.

TEMPORAL

Digital twin technology has an implied temporal component to it, particularly since it deals with physical objects, and physical objects are bound by time. Hardware reliability theory dictates that physical objects will degrade and fatigue over time after periods of usage. Even when idle, physical systems suffer from levels of decay over time. For example, if a car has not been turned on for years, it is likely that the battery will be dead, rust and oxidation has occurred, and perhaps insect or rodent pests have damaged the vehicle. (A friend had his car "totaled" by the insurance company because after years of it lying idle in his barn, he discovered that mice had made nests, clogging components and chewing through all of the wiring.)

A digital twin will not degrade or fatigue over time. Therefore, at some point the real-world entity and digital twin will be in conflict on some level, and synchronization of the two should occur. For example, a metal part could develop hairline fractures after usage that are not represented in the digital twin. This situation might suggest that the digital twin needs to be reworked or maintained to account for this. For example, a physical object at time $t + 1$ will likely be different than at time t. However, the digital twin should be the same at times t and $t + 1$ unless it updates dynamically with feeds from the physical object. Having access to an accurate time stamp for the physical object and digital twin is a trust consideration.

ENVIRONMENT

Digital twin technology has an implied or explicit environmental component that cannot be overlooked. For physical objects, a description of the environmental tolerances or expected usage profiles is needed for many of the "ilities,"[3] particularly interoperability. For example, bricks used to construct buildings are made from a variety of materials; some bricks will break more easily under stress than others, and some bricks are better suited to certain temperatures and climates.[4] This additional expected operational usage information should be reflected in the digital twin. Without this information, it will be difficult to determine if the physical object is "fit for purpose" since purpose implies environment and context. Unknown environmental influences have plagued safety-critical systems and software.

Consider PowerPoint running during a presentation. Usually, the presenter does little more than touch the Page Up or Page Down keys. One could argue that the operational profile for executing PowerPoint during a presentation is twofold: 1) the loaded presentation and 2) the button inputs from the presenter. Whether the presentation goes smoothly (reliably and in a timely manner) is also a function of all of the inputs that PowerPoint is receiving from the disk, memory, and the operating system in real time. If, for example, the presentation gets stuck going from slide x to slide $x + 1$, then something related to "unknown" (phantom-like) environmental influences is probably involved (for example, another process running on the machine at the same time and stealing resources and computing cycles). Accurately defining as many environmental factors as possible is a trust consideration.

MANUFACTURING DEFECTS

Digital twin technology has an interesting relationship to mass production. A digital twin may be used to guide a manufacturing process. For example, a factory that produces light bulbs will have a certain defect rate per thousand bulbs. Not all bulbs produced will be usable, and for those that are usable, there will still be small (possibly microscopic) distinctions between individual bulbs. These small distinctions may impact the lifetime of a specific bulb. The packaging on a set of light bulbs will offer an approximation for how long a bulb will operate before burnout. This facet highlights the idea that a digital twin could not only describe the underlying components of an average bulb but also suggest how it should be manufactured if the representation also details a metric, such as time to burnout. Ensuring that a manufacturing process produces a product with the correct life expectancy based on the information in a digital twin is a trust consideration.

FUNCTIONAL EQUIVALENCE

Digital twin technology requires a means to determine functional equivalence between the digital twin and

the physical object. Without this function, trust is suspect. If the digital twin is an executable specification, then for the inputs that are presented, it should produce the same outputs that the physical object produces for the same input data. If this does not occur, then functional equivalence has not been achieved. This situation could occur for many reasons, such as decay and fatigue, manufacturing variances, or other environmental influences that the physical object experiences during operation but the digital twin does not. Without some assessment of the level of functional equivalence, it is difficult to assert trustworthiness. (Verification and validation can be used to provide evidence of functional equivalence.)

COMPOSABILITY AND COMPLEXITY

There is a trust consideration regarding the size and complexity of the digital twin for its physical object. A digital twin that is too complicated can introduce a composability problem in terms of predicting the trustworthiness of a final composed system from more than one digital twin. Assume that a system has five physical components (real-world entities), and each component has a corresponding digital twin definition. Physically connecting the five components may be straightforward, but composing the five digital twins may not be, particularly if the digital twins contain information such as tolerances and expected operational usages. Standards should be useful to prune extraneous information contained in a digital twin since standards can define required interconnects between components of a domain, enabling the composition to be modeled and tested. One approach might be separating classes of information into categories, such as "essential," "need to know," or "extraneous."

INSTRUMENTATION AND MONITORING

Instrumentation of a digital twin (the ability to provide dashboard information during operations) is a beneficial and unique advantage that digital twin technology offers. While one might not be able to instrument the physical object, one may be able to instrument the digital twin. However, it is not as simple to correctly inject instrumentation and probes into a digital twin as might be expected; much can be learned here from the safety-critical software community. First, a determination of where to inject the probes is necessary.[5] This is not often easy, and it can be more art than science. Second, how many probes to inject is also a consideration. As shown in real-time systems, probes can slow down performance and timing. This may cause a problem for synchronization between the digital twin and physical object. That said, there are ways to reduce this impact by having the probes only collect raw data and not compute internal test results, such as built-in self-tests. Collecting the "right" information from the internal state of an executing digital twin is an expensive and error-prone effort. Finally, the insertion of probes in real systems can sometimes alter the system performance in subtle ways that might not be reflected in the twin.[6] But does the insertion of virtual probes in the digital twin mimic the behavior of the probes in the real system? I don't think so.

> Does the insertion of virtual probes in the digital twin mimic the behavior of the probes in the real system?

HETEROGENEITY OF STANDARDS

The heterogeneity of different formats for digital twins may cause composability problems.[7] If vendors misuse standardized formats for the digital twin definitions of their components, composing digital twin definitions from different component vendors may not be achievable.[8] This is a consideration for trusting composed digital twins.

NONFUNCTIONAL REQUIREMENTS

A trust consideration for systems composed of many components deals with quality attributes often referred to as "ilities." This also applies to digital twin technology. Functional requirements state what a system shall do, negative requirements state what a system shall not do, and nonfunctional requirements (the "ilities") typically state what level of quality the system shall exhibit for both the functional and negative requirements. "ilities" apply to both "things" and the systems they are built into. It is unclear how many "ilities" there are, though examples include availability, composability, compatibility, dependability, discoverability, durability, fault tolerance, flexibility, interoperability, insurability, liability, maintainability, observability, privacy, performance, portability, predictability, probability of failure, readability, reliability, resilience, reachability, safety, scalability, cybersecurity, sustainability, testability, traceability, usability, visibility, and vulnerability.[3] The issue for digital twin technology concerns how many of the nonfunctional requirements can be written for the functional and negative requirements (thus defining the level of quality for what the system should and should not do). The ability to write these nonfunctional requirements will affect the ability to claim the trustworthiness of a composite object.

DIGITAL TWIN ACCURACY

If the accuracy of a digital twin is questionable, or even found to be faulty, then trust is an issue. For software, faulty specifications lead to faulty designs that lead to faulty implementations. In digital twin technology, the degree to which the digital twin is correct is a trust consideration. It begs the question as to whether it might be prudent to have more than one independently created digital twin for a specific physical object. In *N*-version programming,[9] more than one independent software implementation is created for highly critical systems that the software impacts because no single implementation can be assumed to be adequately trustworthy. To address this, each independent implementation is run in parallel, and the outputs from each implementation are sent to a voter, which then decides on the final output that the system receives.

TESTING

The testability of a digital twin refers to measuring how likely an error or defect will be detected during testing. Systems that are less likely to reveal the presence of defects are deemed less testable. Physical objects are testable to different degrees using this definition, though the methods for testing digital twins that are most likely to demonstrate that the digital representation is correct are unclear. One option is to ignore this trust consideration and decide that a digital twin is untestable and, therefore, stands alone as the "oracle" or "gold standard." Moreover, although testing usually involves expected use cases, consideration should also be given for cases of misuse.

CERTIFICATION

Certification usually occurs in two different ways.[10] One type certifies the process used to develop, while the other certifies the final artifact that comes from that process. These two types of certification are distinct.[11-14] For digital twin technology, this means that one could attempt to certify how the digital twin was created or certify the accuracy of the digital twin itself. Certification of a twin will be complicated. For example, the pharmaceutical industry has illuminated the problem of information overload. Most prescription drugs come with warnings concerning who can take them based on gender, age, underlying conditions, negative drug interactions, and other factors. Most drugs also come with disclaimers about negative side effects and when to discontinue use. This information is made available to patients, doctors, pharmacists, and other medical providers.

The problem stems from the vast amount of information known about a drug and the vaster amount of unknown information about a drug at time t that will not be known until time $t + 1$. Further, much of the information is only understandable by medical experts but is vital to determine a drug's fitness for purpose. The trust consideration here for digital twin technology is how much of this information can be provided in a specific digital twin without overloading a user with extraneous information that leads to confusion about how to use the twin or what the twin even represents.

PROPAGATION

One of the greatest trust concerns with any system of systems is how errors and corrupt data propagate (cascade) during execution.[15] Digital twin technology experiences this trust consideration, particularly when different twins representing different physical objects are composed. This may, perhaps, suggest that a digital twin should be packaged with preconditions and postconditions to determine if the output from one digital twin will be acceptable as input to another digital twin.

COUNTERFEITING

It is possible that a digital twin could be tampered with or counterfeited. There are schemes that could be used to protect against this. Digital twin definitions could be hashed and the hash posted to a public webpage; users of a digital twin definition could hash their copy and compare it against the hash on the public webpage. That said, webpages and other similar publicly accessible repositories can be hacked. To enhance trust, one could use a blockchain and post a digital twin definition hash publicly in an immutable data structure (it could never be changed even by malicious attackers). In these ways modifications to digital twin definition files could be discovered. Alternatively, identical copies of a digital twin definition (and related instances) could be stored in separate locations (for example, in offline backups).

In summary, these 14 trust concerns apply to digital twins, but really, to anything virtual. I recommend you think about them in your domain and environment, especially as we enter the metaverse.

REFERENCES

1. "The definition of a digital twin," Digital Twin Consortium, 2020. https://www.digitaltwinconsortium.org/hot-topics/the-definition-of-a-digital-twin.htm
2. J. Voas, P. Mell, and V. Piroumian, "Considerations for digital twin technology and emerging standards," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NISTIR 8356, 2021. [Online]. Available: https://csrc.nist.gov/publications/detail/nistir/8356/draft
3. J. Voas, "Software's secret sauce: The '-ilities' [Software Quality]," *IEEE Softw.*, vol. 21, no. 6, pp. 2–3, 2004, doi: 10.1109/MS.2004.54.
4. *Standard Specification for Building Brick*, ASTM C62 – 17, 2017. [Online]. Available: https://www.astm.org/Standards/C62.htm
5. J. M. Voas and K. W. Miller, "Putting assertions in their place," in *Proc. Int.*

Symp. Softw. Reliability Eng., Monterey, CA, USA, 1994, pp 152–157. doi: 10.1109/ISSRE.1994.341367.
6. P. Laplante, "Heisenberg uncertainty," *ACM SIGSOFT Softw. Eng. Notes*, vol. 15, no. 5, pp. 21–22, 1990, doi: 10.1145/101328.101333.
7. J. M. Voas, "Networks of 'Things'," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NIST Special Publication (SP) 800-183, 2016.
8. J. M. Voas and P. Laplante, "Standards confusion and harmonization," *Computer*, vol. 40, no. 7, pp. 94–96, 2007, doi: 10.1109/MC.2007.252.
9. L. Chen and A. Avizienis, "N-version programming: A fault-tolerance approach to reliability of software operation," in *Proc. 8th Int. Symp. Fault-Tolerant Comput.*, 1978, pp. 3–9.
10. J. Voas, "The software quality certification triangle," *Crosstalk*, vol. 11, no. 11, pp. 12–14, 1998.
11. J. M. Voas and G. Hurlburt, "Third party software's trust quagmire," *Computer*, vol. 48, no. 12, pp. 80–87, 2015, doi: 10.1109/MC.2015.372.
12. J. Voas, "Toward a usage-based software certification process," *Computer*, vol. 33, no. 8, pp. 32–37, 2000, doi: 10.1109/2.863965.
13. J. Voas and P. Laplante, "The IoT blame game," *Computer*, vol. 50, no. 6, pp. 69–73, 2017, doi: 10.1109/MC.2017.169.
14. J. Voas and P. Laplante, "IoT's certification quagmire," *Computer*, vol. 51, no. 4, pp. 86–89, 2018, doi: 10.1109/MC.2018.2141036.
15. J. Voas, "Error propagation analysis for COTS systems," *IEEE Comput. Control Eng. J.*, vol. 8, no. 6, pp. 269–227, 1997, doi: 10.1049/cce:19970607.

PHIL LAPLANTE is a professor of software and systems engineering at The Pennsylvania State University, Malvern, Pennsylvania, 19355, USA, a Fellow of IEEE, and an associate editor in chief of *Computer*. Contact him at plaplante@psu.edu.

Digital Object Identifier 10.1109/MC.2022.3178652