EXAMINING MODEL QUALITIES AND THEIR IMPACT ON DIGITAL TWINS

Bentley James Oakes

Université de Montréal
2920 chemin de la Tour, Montréal, CANADA
bentley.oakes@umontreal.ca

Cláudio Gomes
Peter Gorm Larsen

Aarhus University
Åbogade 34, Aarhus N, DENMARK
{claudio.gomes,pgl}@ece.au.dk

Joachim Denil

Flanders Make and University of Antwerp
Groenenborgerlaan 171, Antwerp, BELGIUM
joachim.denil@uantwerpen.be

Julien Deantoni

Université Côte d'Azur - Sophia Antipolis
2000 route des Lucioles, Sophia Antipolis, FRANCE
julien.deantoni@univ-cotedazur.fr

João Cambeiro

Université Côte d'Azur - Sophia Antipolis
2000 route des Lucioles, Sophia Antipolis, FRANCE
NOVA University
Largo da Torre, Monte da Caparica, PORTUGAL
joao.cambeiro@univ-cotedazur.fr

John Fitzgerald

Newcastle University,
Newcastle upon Tyne, UK
john.fitzgerald@ncl.ac.uk

ABSTRACT

Digital Twins (DTs) are built using modelling and simulation techniques in complex domains such as cyber-physical systems. However, further formal investigation is required for how a DT and the services it provides relate to the qualities of the models used by a service. Specifically, this article examines when a DT service can be said to have the qualities of *relevant*, *verifiable*, *substitutable*, and *faithful* based on the results of checking *properties* in comparison to the actual system. Using an incubator system as our running example, we show how a DT service relies on multiple models, present the consequences when these qualities are violated, and discuss strategies for adapting models to ensure these qualities.

Keywords: digital twins, verification, model quality, fidelity, substitutability, cyber-physical systems.

1 INTRODUCTION

Digital twins (DTs) are an emerging topic in industry and academia as they leverage the principles of modelling and simulation to monitor, reason about, control, and optimize, practices and systems. As a definition: "A [DT] is a virtual replica of physical assets, processes, people, places, systems or devices created and maintained in order to answer questions about its physical counterpart (the *physical twin* (PT))." (Fitzgerald,

Larsen, and Pierce 2019). Because the extent of "virtual replica" in the DT definition is unclear, we prefer to think of a DT as being comprised of a number of *DT services*, which are then defined unambiguously (see section 2). DTs are highly relevant in the area of *cyber-physical systems* (CPSs) including self-driving cars, industrial machinery, and smart grid technologies.This wide range of uses comes with many challenges stemming from the complex behaviour of heterogeneous CPS components interacting among themselves and their environment. DTs assist in managing this complexity of CPSs by offering a virtual representation of the PT which can be simulated and verified. As data flows from the PT to the DT, it can be analyzed or used to predict the future behaviour of the PT. *DT services* can then inform humans (e.g., dashboards, visualizations) or even perform supervisory control of the PT (e.g., safety shutoff, optimization).

Motivation Two characteristics of DTs motivate our contribution (see section 2 for examples): a) DT services rely on models of the PT; and b) DTs are likely to evolve and provide new services, just as the PT and its environment are likely to change. These characteristics raise the issue that the *DT must evolve in a dependable and reliable manner alongside the PT* to increase the value of the PT, while ensuring that *model qualities are preserved when deployed in the context of services*.

Contribution & Structure This article formally defines four distinct qualities of a model (*relevance*, *verifiability*, *substitutability*, and *fidelity*), and explores how these qualities impact the services that depend on the model. Our definitions build upon (Denil et al. 2017, Van Acker et al. 2021, Biglari and Denil 2022) and (Cambeiro, Deantoni, and Amaral 2021), but we take a higher-level view to discuss the four inter-related qualities in a CPS DT context. In particular, we: introduce the incubator running example (section 2); define these qualities through the comparison of property satisfaction on the PT and on a model (section 3); show how these property satisfaction results relate to the models used by DT services (section 4); show examples of how quality degradation will cause DT services to be unreliable (*not dependable*) when they are not respected (section 4); and discuss adaptation strategies to preserve model and service qualities (section 5).

2 DIGITAL TWINS AND A RUNNING EXAMPLE

The concept of DT originated in the context of Product Lifecycle Management (PLM) (Grieves and Vickers 2017). However, in this article we consider the DT in the context of the operation of CPS which are systems such as automobiles and robotics containing both computational component(s) and physical component(s).

Fundamentally, DTs are the virtual support systems of the PT, including representations of the physical objects of the PT, but also representations of processes, the environment influencing the PT, or historical information, as well as offering decision support functionalities (Jones et al. 2020). The representations are supported by *models* of different PT aspects describing a variety of viewpoints. The DT also retrieves *data* about the PT, either in a *streaming* manner or from a *historical* data repository (Oakes et al. 2021).

The DT provides value by reasoning about this combination of models and data, and providing useful information through *services*. Services allow stakeholders to gain information about the PT in the past, present, and future in a variety of scenarios without affecting PT operation. For example, a DT service could determine if the PT will enter an unsafe state in a different environment. If a DT offers services which performs *actions* (such as PT control and adaptation) (Oakes et al. 2023), then by the classification of (Kritzinger et al. 2018) it is a *digital twin*. With no automatic actions, the DT is a *digital shadow*.

Thus a DT provides value by providing "useful/dependable" information through its services. However, it is important to ensure that as the set of DT services changes, or the PT evolves (new environment, degradation, etc.) that this information is still useful. In section 4 we discuss how the model qualities introduced in the next section relate to the models which a service relies on. These qualities, if not respected, will render a service "useless" leading to incorrect and potentially damaging misinformation.

Incubator Example The incubator is composed of an insulated box with a heating element to maintain a stable temperature for objects placed inside (Feng et al. 2021, Feng et al. 2022). A conceptual diagram is found in fig. 1 based on the DT characteristics of (Oakes et al. 2021, Oakes et al. 2023). The right-hand side shows the incubator system and environment. The PT contains temperature *sensors* and *actuators* for the heater and fan. The operator can also influence the incubator system through manual operation of the lid. From the DT to the incubator, commands are sent from the DT to the actuators and a dashboard is displayed to the operator. From the incubator to the DT, *real-time* temperature readings are sent.

The left-hand side of fig. 1 is a representation of the information flow within the DT. On the bottom are the *models and data*, consisting of temperature readings, a differential equation model for temperature propagation, and a Kalman filter (a linear quadratic estimation algorithm) model used for estimating the hidden states of the incubator. The *enablers* on the middle layer are computational components utilizing these models and data to support the services. On the top layer are the *services* that the DT provides as an interface to the user and the actuators. Note that we treat DT services as interfaces that expose the information calculated by enablers, as well as receiving input from users through an API/GUI. That is, any significant computation is performed by enablers and models. This is important when we discuss the *verifiability* of properties in section 4. This collection of services, enablers, and models/data is termed a "constellation" by (Oakes et al. 2023) because a web of data and control dependencies can be represented.

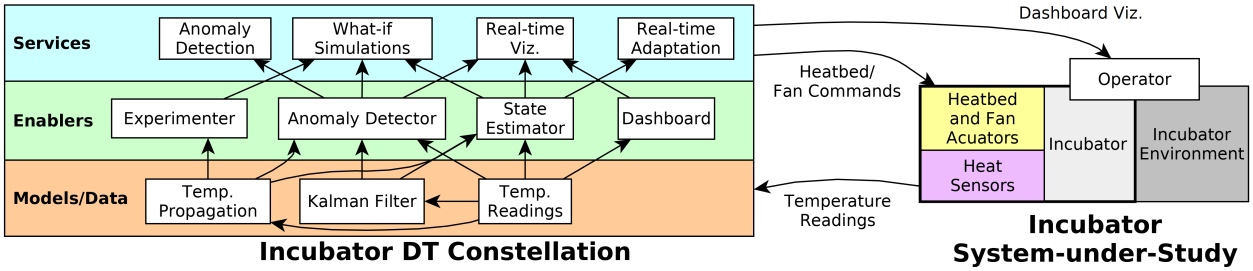


Figure 1: A conceptual representation of the incubator system and its digital twin.

DT and PT Evolution Throughout the DT's life, new services are added, with new data dependencies between elements, and the models, data, and enablers, are replaced by new versions. For example, a *visualization* enabler could be added to demonstrate the transfer of heat by components, long after the incubator DT has been in operation. New models of the incubator operating with the lid opened could also be added to support new services that perform anomaly detection. As an example of a PT evolving, products such as tempeh (a food made from fermented soybeans) shift from being heat absorbers to heat producers in incubation (Feng et al. 2022). Different models may thus have to be utilized to reflect how various incubated objects affect the temperature.

3 THE RELATIONSHIP BETWEEN A SYSTEM AND ITS MODELS

As a DT is composed of multiple models which provide information to a service, it is crucial to first discuss the relationship of a system to the models which represent that system. In particular, this section introduces the basic concepts of a *system*, its *context*, *properties*, and our concept of *models*. We also formalize the model qualities of *relevancy*, *verifiability*, *substitutability*, and *fidelity* with respect to the system.

3.1 Systems, Their Contexts, and Properties

A *system* can be defined as "a combination of interacting elements organized to achieve one or more stated purposes"(IEEE 2015). For CPSs, this may include the physical device itself, natural phenomena surround-

ing the CPS, or less tangible elements such as information about the CPS. The elements that the system interacts with but are not part of the system itself form the *environment*. The *system boundary* between the system and its context must be carefully chosen through *systems engineering* as discussed by (Zeigler 1976). Most relevant for our purposes is that this boundary is often chosen based on the system purpose.

A system's *purpose* relates to its *behavior* (its interactions with its environment). The set of all possible interactions from the environment to the system is denoted by the system *context*. Note that we focus on interactions from the environment to the system because we apply these properties to DTs, whose role is to account for novel effects of the environment on the system. We focus here on system *properties* coming from functional and non-functional requirements, where these properties are conditions on the observable behaviour of the system, and thus encode the purpose of the system. These properties should be *satisfiable* such that we can determine whether the system satisfies a property in a certain context or not, taking into account any possible uncertainty in the *measurements* required for checking the property.

Notation. *Let S denote the system, and let P_S denote the set of properties that S's behavior should satisfy in context C_S. Then we use the notation $[\![S]\!]_{C_S}$ to represent the behavior of S produced under context C_S, and we write $[\![S]\!]_{C_S} \vDash p$ to state that $[\![S]\!]_{C_S}$ satisfies the property $p \in P_S$.*

Example 1. *The incubator system includes the incubator itself, the object(s) placed inside, the heating element, and the controller. The environment includes the air in the surrounding room, the table which the incubator is placed upon, and so on. The purpose of the incubator system is to keep the temperature stable around a given value, regardless of the ambient temperature outside the box. The system boundary is the insulated box itself. Since the box is not perfectly insulated, more energy will be used by the incubator heating element to retain the target temperature when the room is colder. Example properties include: the incubator system takes maximum 10 seconds to warm up the air from* 20°C *to* 30°C*, the incubator temperature remains within a minimum and maximum at all times, the incubator controller takes maximum three seconds to react to variations, etc. The context includes all possible room temperature profiles which the incubator is subjected to, as well as all possible usages of the incubator.*

3.2 Models and their Qualities

Following (Kühne 2005, Stachowiak 1973), a *model* needs to be based on the system, reflect only a relevant subset of its properties, and can be used in place of the original, for some pre-defined purpose and within a particular context. A model is created for the purpose of experimenting with it, and with a set of *properties of interest* and context in mind (Cellier 1991, Zeigler 1976). For our purposes, the property satisfaction results provided by performing an experiment on a model (in its experimental context) should be *faithful* to that of determining the property satisfaction on the real system (in its context) as discussed by (Denil et al. 2017). This notion of "faithfulness" is made more precise in section 3.6 as *fidelity*.

Notation. *We use the notation $[\![M]\!]_{C_M}$ to represent the behavior of the virtual experiment on M performed under context C_M, and we write $[\![M]\!]_{C_M} \vDash p$ to state that $[\![M]\!]_{C_M}$ satisfies the property $p \in P_M$, where P_M denotes the properties of interest that M's behavior should satisfy.*

Example 2. *The incubator system is comprised of a controller software, and a plant consisting of the heating element, the air inside the box, and the box wall. A model for calculating the average temperature evolution (ignoring objects within the box) is provided by (Feng et al. 2021):*

$$\begin{aligned} \dot{T}_{heater} &= \frac{1}{C_{heater}} \cdot \left(V \cdot I \cdot \Delta t - G_{heater} \cdot (T_{heater} - T_{boxair})\right) \\ \dot{T}_{boxair} &= \frac{1}{C_{air}} \left(G_{heater} \cdot (T_{heater} - T_{boxair}) - G_{box} \cdot (T_{boxair} - T_{room})\right) \end{aligned} \tag{1}$$

Relevant to our discussion is that the parameters in eq. (1), such as the entropy constant C_{air}, must be given a concrete value for eq. (1) to form a model, and that each new set of parameters will form a distinct model.

In the following, we explore model qualities to determine if such a model is appropriate for the current system's context (*relevancy*), whether it can be used to check if properties are verified (*verifiability*), whether the model can stand in for the real system (*substitutability*), and whether the model's behavior will closely match the real system behavior (*fidelity*). As in (Denil et al. 2017), a model not exhibiting these qualities is said to be *invalid*.

Remark 1. *The conditions for each quality should be read and understood while assuming that the other qualities of the model hold. All qualities must hold in practice, so we do not lose generality.*

3.3 Purpose & Relevance

The set of properties P_M encodes the *purpose* that the model was built for. The notion of *relevance* examines if the model is indeed based on the system. Essentially this is whether the context of the model covers the context of the system, which is not trivial in practice. For instance, (Spiegel, Reynolds, and Brogan 2005) describe an experiment where experts in physics are asked to identify the implicit assumptions in a simple model of a particle moving in a viscous medium. These assumptions represent situations in which the model would not be a predictor of the system's behavior. No expert was able to identify all the 29 assumptions identified by their combined expertise.

Equation (2) represents our formalization of *relevancy* of M w.r.t S: a model is relevant if, considering the context of the system restricted to the one of the model (denoted by $C_S|_{C_M}$), for each of the system's interactions, there exists the same interaction in the model, and this model interaction encompasses its system analogue:

$$relevant(M,S) \stackrel{def}{=} \forall i_S \in C_S|_{C_M}, \exists i_M \in C_M \text{ such that } i_M \supseteq i_S \tag{2}$$

Example 3. *For example, the equations in eq.* (1) *are constructed with the assumption that the incubator is situated within a room containing air. If this incubator is thrown into the ocean, or deployed in space, then the water or vacuum will interact with the incubator elements in a way that the equations were not constructed to account for. These equations will not be relevant and they will not produce the same behavior as the incubator. When the incubator is in a normal, air-filled environment, the equations are relevant.*

3.4 Verifiability

Verification is the act of checking that a model satisfies the properties of interest. More formally, for a given property p_i, *verifiability* of p_i means the ability to conclude whether $[\![M]\!]_{C_M} \vDash p_i$ is true or false (noted $[\![M]\!]_{C_M} \stackrel{?}{\vDash} p_i$). A model is *verifiable*, when it is possible to conclude about satisfaction for all properties $p \in P_M$.

$$verifiable(M) \stackrel{def}{=} \forall p_i \in P_M, [\![M]\!]_{C_M} \stackrel{?}{\vDash} p_i \tag{3}$$

Remark 2. *The short formalization in Equation* (3) *abstracts many concrete details. (i) The act of computing the behavior of the model ($[\![M]\!]_{C_M}$) may involve approximations in practice. For example, the discretization of the continuous time in Equation* (1) *(Cellier and Kofman 2006). (ii) The satisfaction relation between the behavior and the property may be a three-valued logic. For example, consider a property where the temperature of the incubator should always be below* 80°C. *The approximation error of the model may be* ±2°C. *Thus, there is a range where the property satisfaction is inconclusive and consequently verifiability does not hold. (iii) The satisfaction relation may also take the form of a real-value or probabilistic satisfaction, such as determining whether the value is often within specified bounds. (iv) Finally, the property may be undecidable with respect to the model, such as checking stability for highly non-linear systems.*

3.5 Substitutability

The quality of *substitutability* is implicit in the model's ability to "reflect a subset of the system's properties". Reflection means that there is a bi-directional implication between the properties satisfied by the behavior of the system and the ones satisfied by the behavior of the virtual experiment of the model. That is, the substitutability of a system S by a model M holds when:

$$substitutability(S,M) \stackrel{def}{=} \forall p \in P_M, (\llbracket M \rrbracket_{C_M} \vDash p) \Leftrightarrow (\llbracket S \rrbracket_{C_S} \vDash p) \tag{4}$$

Example 4. *Consider the model presented in example 2 and the property "the incubator temperature remains within a minimum and maximum at all times". There is substitutability if both the behavior of the system and the behavior of the virtual experimentation of the model agree on the satisfaction of this property (i.e., if not satisfied by the system then not satisfied by the virtual experiment and vice versa).*

3.6 Fidelity

Fidelity is a concept that is widely used, but hard to define formally. For us, it represents the bridge between how a model is typically used in practice and the substitutability of a model. We thus assume that there is a measure of the fidelity $g_{p_i}(\llbracket M \rrbracket_{C_M}, \llbracket S \rrbracket_{C_S})$. This real-valued function g_{p_i} depends on each property p_i and establishes a real-valued distance between the observed behavior w.r.t. p_i for both the model and the system. For the operational details for the calculation of the deviation and determining tolerances, we point the reader to recent work by (Muñoz et al. 2022) and (Biglari and Denil 2022).

With this function g, we can formally define what it means for a model to have *sufficient fidelity*. Given a property-dependent small $\varepsilon_i \in \mathbb{R}$, a model M has *sufficient fidelity* w.r.t. to a property p_i when its behavior traces are sufficiently close to the behavior traces of the system S, so that substitutability holds:

$$g_{p_i}(\llbracket M \rrbracket_{C_M}, \llbracket S \rrbracket_{C_S}) < \varepsilon_i \implies (\llbracket M \rrbracket_{C_M} \vDash p_i) \Leftrightarrow (\llbracket S \rrbracket_{C_S} \vDash p_i) \tag{5}$$

Equation (5) attempts to embody the engineering practice: a person builds a model from prior knowledge of how the model's behavior matches (within some tolerance ε_i) the real system $\llbracket S \rrbracket_{C_S}$, or some other reference behavior. Presumably, this matching implies that the model can be used as a substitute (or predictor) of property satisfaction for the real system (or a reference behavior).

Remark 3. *The verifiability of the model affects the fidelity as well, but for clarity we assume in these definitions that it does not (recall remark 1). For instance, for a model where temperature reachability cannot be computed (verifiability), this means that the distance between the reachable states and the real system data is also impossible to assess (fidelity).*

Example 5. *Recall example 2. If we observe the distance between these equation's behavior and the equivalent experiments on the incubator (in a particular context) remains within some small constant, then substitutability holds. This quality is essential for proving safety properties such as ensuring the incubator remains within the correct operating range (Wright, Gomes, and Woodcock 2022).*

3.7 Example Relating Model Qualities

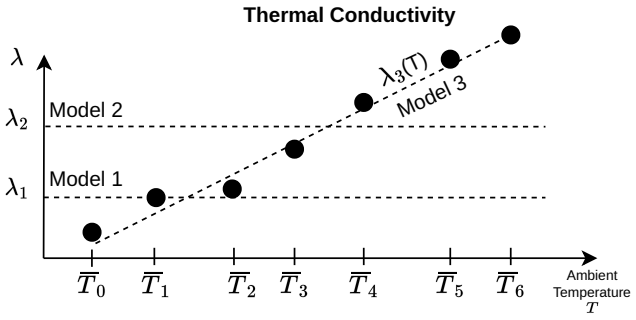
The following example, presented within the scope of the incubator, demonstrates the relationships between the different model qualities introduced before. Thermal conductivity is an important factor in our example models to correlate the temperature differences inside and outside the incubator. The heat transfer rate is Fourier's law $q = -\lambda(T)\frac{\Delta T}{d}$ where q denotes the heat flux, or rate of heat transfer per unit of area in

units $\frac{W}{m^2}$, $\lambda(T)$ is the thermal conductivity (which depends on the ambient temperature), $\Delta T = T_1 - T_2$ represents the temperature difference, and d represents the thickness of the material. One way to estimate λ in Fourier's law is a *steady-state method*, consisting of applying a known heat flux q to a sample with a surface area A and thickness d. Here we construct three different models which are instantiations of Fourier's law with different specifications of λ. These models are constructed by performing seven experiments with the incubator situated in various ambient temperatures.

Model 1 $\lambda(T) = \lambda_1$ is a constant, taken from a context where the ambient temperature was $\bar{T}_1$.

Model 2 $\lambda(T) = \lambda_2$, with λ_2 taken from the average thermal conductivities of two experiments, where the ambient temperature was $\bar{T}_3$ and $\bar{T}_4$.

Model 3 $\lambda(T) = \lambda_3(T)$ is a linear function of the ambient temperature, regressed from all experiments.



	Context	Fidelity
Model 1	$\{\bar{T}_1\}$	High
Model 2	$\{\bar{T}_3, \bar{T}_4\}$	Low
Model 3	$\{\bar{T}_1, \bar{T}_2, \bar{T}_3, \bar{T}_4, \bar{T}_5, \bar{T}_6\}$	Medium

Figure 2: Relation between insulator temperature and thermal conductivity.

Figure 2 shows the results of the seven experiments performed at various ambient temperatures. The dashed lines indicate the instantiation of λ for each model. The right-hand side of Figure 2 indicates the *context* for each of the three models, with a fidelity value presented qualitatively for simplicity. A quantitative fidelity would be the absolute difference between the dashed model and the nearest empirical result.

Relevance Assume that the context of the system C_S consists of all temperatures between $\bar{T}_0$ and $\bar{T}_6$. That is, the incubator will be placed in each of these ambient temperatures. In this case, the only relevant model, according to eq. (2) is Model 3.

Substitutability & Fidelity While it does not make sense to talk about these qualities for models that are not relevant, we do so for the sake of comparing them as in (Van Mierlo et al. 2020). Model 1 is has sufficient fidelity when the ambient temperature surrounding the incubator wall is *close enough* to $\bar{T}_1$. Model 2 has a larger context compared to Model 1, however, its fidelity is lower, since we expect its predictions to be farther from the real experiment data when the temperature surrounding the incubator wall is *far from* $\bar{T}_3$ and $\bar{T}_4$. As expected, Model 3 was built from system experiments covering a wider range of ambient temperatures, and therefore the model itself can retain a higher level of fidelity for varying levels of room temperature. It is important to note that when a model is used outside of its context, it could still have sufficient *fidelity*. However, as it is not *relevant*, then there might exist a situation in the system's context that is not in the model context, where such a situation may break the model's *substitutability*.

4 THE RELATIONSHIP BETWEEN A SYSTEM AND ITS DIGITAL TWINS

As the DT and the PT evolve, the DT models and services may have to be reconfigured to ensure that they are still *relevant*, *verifiable*, *substitutable*, and *faithful*. Here we break down what these qualities mean for DT services using our incubator example, and we examine the issues that can arise if these qualities are not maintained when the DT or system evolves.

4.1 Services Relying on Models

We introduce notation for how a service relies on its models, indirectly through its enablers as defined in (Oakes et al. 2023). Let a service be $v \in V$. Then the models that the service relies on will be defined as $m \in M_v$. That is, the models $m \in M_v$ provide information to the service v.

We clarify three notes here. i) our characterization is that a *service merely reports the results of computations* performed within a model. That is, the service does not contain significant computation such as value averaging or error handling, and instead services simply pass along the information from the models. ii) we assume that the *outputs provided by a model are important* to the operation of a service. If disabling the computation of a model would not majorly impact the service, then there is not a strong reliance from the service on the model. This is crucial as it only makes sense to examine these qualities for models where their output is critical to the service's purpose. iii) we also make the assumption that the data provided to the DT by the PT is accurate and timely. That is, that the data quality is sufficient such that the operation of the model and the service are not affected. This assumption may not hold on a real system, as sensor precision, sensor and signal noise, and transmission delays may all impact the data used for model predictions

Incubator Example The incubator controller service (*adaptation service*) relies on an accurate model of how the incubator *temperature propagates* as introduced in eq. (1). Thus the service is $v_{adaptation}$ and the model is $m_{tempprop} \in M_{v_{adaptation}}$. This dependency is also captured graphically in fig. 1 by arrows from *temperature propagation model* to the *state estimator enabler*, and then to the *adaptation service*.

We summarize below what happens to a service when these qualities are not respected for models. The following sections will discuss each of the model qualities for services and the models they rely upon. Section 5 then discusses the order of checking these qualities and adapting the models.

- *Relevancy*: If any model is *not relevant* for the system's context, then the service is also *not relevant* (section 4.2).
- *Verifiability*: If a property *is not verifiable* for at least one model for the service, then it *cannot be verified* for the service (section 4.3).
- *Substitutability*: If a property *cannot be satisfied* for at least one model for the service, or is *not satisfied* by any model, then the service is *not substitutable* for the system (section 4.3).
- *Fidelity*: If a property's value produced by any model is *insufficiently faithful* to the system, then the service *most likely* is also insufficiently faithful (section 4.4).

4.2 Relevancy

Recall that *relevancy* for the model-system relation was whether the model context was equal to or larger than the (relevant) system context. In our characterization of DTs, a service relies on many models to perform computations and provide PT information. Therefore, our relevancy definition for a DT service is:

Definition 1. *For a DT service to be relevant (with regards to the PT), all models relied upon by that service must be relevant (with regards to the PT) as in eq.* (2)*. Therefore, a given service v is relevant iff:*

$$\forall m_j \in M_v : \forall i_S \in C_S|_{C_{M_j}}, \exists i_{M_j} \in C_{M_j} \text{ such that } i_{M_j} \supseteq i_S \tag{6}$$

Incubator Example Section 3.7 discussed the context ranges for the prediction of the heat transfer within the incubator. Consider the *real-time adaptation service* which adjusts the operation of the heat-bed and fan based on the simulated heat transfer provided by those models. If a *non-relevant* model, that was calibrated for a different ambient temperature than the system is placed into, is used as basis to optimize the heat-

bed controller service, the consequences could be severe. Extreme examples include underheating of the incubated object (fatal for an incubated chicken egg) or overheating of the incubator (a safety hazard).

Practically, the relevancy information for each model in a service should be explicitly recorded in a *validity frame* (Denil et al. 2017). In these, a set of experiments provides the *validity/context conditions* (Van Mierlo et al. 2020) such as those specified in fig. 2. These conditions can then be checked against the (measured) context of the system to determine if any model is drifting out of relevancy.

4.3 Verifiability & Substitutability

Verifiability depends on whether the models in the DT that the service relies upon can be checked for satisfaction of the properties of interest.

Definition 2. *For a DT service to be verifiable (with regards to a property), at least one model relied upon by that service must be able to satisfy (or not satisfy) that property.*

Furthermore, our definition of *substitutability* builds on this verifiability definition. A *service v is substitutable for the real system w.r.t. property p* iff:

$$\exists m_i \in M_v : \quad (\llbracket m_i \rrbracket_{C_M} \not\models p \wedge \llbracket S \rrbracket_{C_S} \not\models p) \quad \vee \quad \Big((\llbracket m_i \rrbracket_{C_M} \models p \wedge \nexists m_j \in M_v : \llbracket m_j \rrbracket_{C_M} \not\models p) \wedge \llbracket S \rrbracket_{C_S} \models p\Big) \tag{7}$$

The first term in eq. (7) examines whether there is a model where a property is not satisfied, just like the system. In the second term, if there are conflicting satisfaction answers where some models suggest that the property is satisfied and other models suggest it is not, we make the safe assumption that the service cannot be substituted for this property and adaptation is required to correct this inconsistency.

Incubator Example Consider the property to be verified on the dashboard service that the *incubator heat-bed shall never have a temperature* $> 80°$C. The models involved in the dashboard must be examined to see if the property holds. If tools and techniques are not able to check the satisfaction of this property, it is not *verifiable*. To check the *substitutability* of the service for this property, the *experimenter enabler* of the incubator DT could simulate the incubator forward in time using a Monte-Carlo approach to determine if the property is satisfied or not. A more formal approach is to consider reachability analysis of the system's dynamics, such as using temporal logic in (Wright, Gomes, and Woodcock 2022). If these approaches do not agree on the satisfaction of the temperature constraint, then the property cannot be properly verified for the service. The discrepancy will need to be further investigated to determine its cause, such as an over- or under-approximation in a model.

In general, the ability to check these conditions depends on the formalism of the property and the model, and the tooling required. First, it must be possible to produce a behavior trace for the system. Second, the behavior trace must then be examined to determine if the property is satisfied or not, which may be expensive or difficult to perform in some formalisms without significant effort.

4.4 Fidelity

Fidelity is a common term when discussing DTs, such as by (Jones et al. 2020): "the higher the fidelity, the closer the virtual and physical twins are aligned". This oft-expressed notion is that fidelity is a measure of the DT itself such that the DT can be said to be high or low fidelity with respect to the PT. However, in our view, this is too general in a modelling and simulation context. Instead, we focus on the level of fidelity *per property* and *per service*. For example, representing the exact temperature of the heat-bed is more important for a *what-if simulator* (as the system dynamics may be sensitive to this value) than a *visual dashboard* (where the temperature may be truncated to just whole degrees).

Definition 3. *For a service v_j, let $f_{p_i} = g(p_i, [\![m_k]\!]_{C_{m_k}}, [\![S]\!]_{C_S})$ and $\varepsilon_{ij} \in \mathbb{R}$ be a given tolerance dependent on the property and service. Then v_j has sufficient fidelity with respect to property p_i iff:*

$$\forall m_k \in M_{v_j} : (f_{p_i} < \varepsilon_{ij}) \implies \big(([\![m_k]\!]_{C_M} \vDash p_{ij}) \Leftrightarrow ([\![S]\!]_{C_S} \vDash p_i)\big) \tag{8}$$

As can be seen here, fidelity depends on all models used in the service providing sufficiently faithful results.

Incubator Example Recall the models in section 3.7, which were stated to have higher or lower fidelity w.r.t. the rate of heat transfer. Depending on the service which uses these models, the level of fidelity may be sufficient or insufficient. For example, a dashboard for the human operator may not need precise behavior w.r.t. heat transfer. Adaptative abstraction/approximation to maintain fidelity is discussed in section 5.

5 DISCUSSION: QUALITY PRIORITIES AND ADAPTATION STRATEGIES

When a DT or the PT evolves, the models within a DT should be examined to ensure they are still useful. Thus we present here an ordering to ensure model qualities, which is important for safety-critical CPSs where there is limited time or computational resources to raise an alarm before an incident occurs.

Checking *verifiability* is essential for correct DT operation. If models cannot be checked for the properties of interest, then it will not be possible for them to provide *substitutable* results for the service.

Substitutability is crucial for modelling and simulation purposes, and the satisfaction for properties should be examined next using approaches such as in (Wright, Gomes, and Woodcock 2022). However in many cases properties may not be able to be checked for satisfaction. For example, there could be a large possibility space which cannot be formally checked for property satisfaction. Note that formal approaches may take a longer amount of time, and quick or approximate satisfaction checks may be preferred.

Next, the *relevance* of a model must be examined, to ensure that the models within a DT are relevant. As mentioned, non-relevant models can lead to damage or injury in safety-critical systems. Approaches such as validity frames can assist in monitoring this quality (Van Acker et al. 2021, Biglari and Denil 2022). Note that it is still important to have relevance if the properties of interest are verified. This is because the risk is higher that the results will not be correct with slight perturbations of the system. That is, the dependability/reliability of the DT is much lower if the models used by DT services are not relevant.

Fidelity is only important once *verifiable*, *substitutable*, and *relevant* models are being used. Here, the criterion is whether the fidelity is sufficient for the service to continue to reflect satisfaction of the system's properties of interest. Again, at some point, damage or injury may be caused if the fidelity is insufficient. In other cases, such as in visualization of a system, low-fidelity may be low-impact or even desired to lower resource requirements. The fidelity of the services and models should be improved when possible to ensure that sufficiently-accurate values are relied on (Cambeiro, Deantoni, and Amaral 2021).

Adaptation strategies are required when a model cannot provide satisfaction results, provides incorrect verification results, is not relevant, or is insufficiently faithful. One strategy is to simply *switch out* the model for a different one, potentially requiring a complicated initialization procedure (Biglari and Denil 2022). Alternatively, the model parameters may simply be *recalibrated* to improve the model's results as in (Feng et al. 2021). These adaptations are required to ensure that DT services continue to provide relevant, verifiable, substitutable, and faithful results while still balancing computational demands.

6 CONCLUSION

This article has explored the concepts of when a DT service provides dependable information about the system, through the formalized qualities of *relevancy*, *verifiability*, *substitutability*, and *fidelity*. These qual-

ities are related to the running example of an incubator CPS, where violations of these qualities leads to incorrect results provided by the services. The monitoring and adaptation of these qualities is also briefly discussed. We present one running example here due to space constraints. However, we have cross-checked these concepts against other DT case studies that we are involved in.

The current major limitation is that the definitions we provide here require further effort to be realized in a DT framework implementation. Our goal is to inform future implementations, but these concepts will have to be refined and possibly restricted for computational tractability. Also, we do not consider composition of models and properties that span multiple models. Future work will focus on deepening the formalization of these qualities to handle these considerations. We will also further develop self-adaptation frameworks such that DT services are able to detect violations of these qualities during system operation and dynamically configure their models.

ACKNOWLEDGEMENTS

We acknowledge the Poul Due Jensen Foundation for funding the project *Digital Twins for Cyber-Physical Systems* (DiT4CPS).

REFERENCES

Biglari, R., and J. Denil. 2022. "Model Validity and Tolerance Quantification for Real-Time Adaptive Approximation". In *25th International Conference on Model Driven Engineering Languages and Systems: Companion (MODELS-C) Proceedings*, pp. 668–676.

Cambeiro, J., J. Deantoni, and V. Amaral. 2021. "Supporting the Engineering of Multi-Fidelity Simulation Units With Simulation Goals". In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 317–321. Institute of Electrical and Electronics Engineers, Inc.

Cellier, F. E. 1991. *Continuous System Modeling*. Springer Science & Business Media.

Cellier, F. E., and E. Kofman. 2006. *Continuous System Simulation*. Springer.

Denil, J., S. Klikovits, P. J. Mosterman, A. Vallecillo, and H. Vangheluwe. 2017, April. "The Experiment Model and Validity Frame in M&S". In *Proceedings of the Symposium on Theory of Modeling & Simulation*, TMS/DEVS '17, pp. 1–12. San Diego, CA, USA, Society for Computer Simulation International.

Feng, H., C. Gomes, S. Gil, P. H. Mikkelsen, D. Tola, P. G. Larsen, and M. Sandberg. 2022, July. "Integration Of The Mape-K Loop In Digital Twins". In *2022 Annual Modeling and Simulation Conference (ANNSIM)*, Institute of Electrical and Electronics Engineers, Inc.

Feng, H., C. Gomes, C. Thule, K. Lausdahl, A. Iosifidis, and P. G. Larsen. 2021, July. "Introduction to Digital Twin Engineering". In *2021 Annual Modeling and Simulation Conference (ANNSIM)*, Institute of Electrical and Electronics Engineers, Inc.

Feng, H., C. Gomes, C. Thule, K. Lausdahl, M. Sandberg, and P. G. Larsen. 2021, February. "The Incubator Case Study for Digital Twin Engineering". *arXiv:2102.10390 [cs, eess.SY]*.

Fitzgerald, J., P. G. Larsen, and K. Pierce. 2019. "Multi-Modelling and Co-Simulation in the Engineering of Cyber-Physical Systems: Towards the Digital Twin". In *From software engineering to formal methods and tools, and back*, pp. 40–55. Springer.

Grieves, M., and J. Vickers. 2017, August. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". In *Transdisciplinary perspectives on complex systems*, pp. 85–113. Springer.

IEEE 2015. *International Standard ISO/IEC/IEEE 15288:2015(E), Systems and software engineering — System life cycle processes*. ISO/IEC and IEEE Computer Society.

Jones, D., C. Snider, A. Nassehi, J. Yon, and B. Hicks. 2020. "Characterising the Digital Twin: A Systematic Literature Review". *CIRP Journal of Manufacturing Science and Technology* vol. 29, pp. 36–52.

Kritzinger, W., M. Karner, G. Traar, J. Henjes, and W. Sihn. 2018. "Digital Twin in Manufacturing: A Categorica Literature Review and Classification". *IFAC-PapersOnLine* vol. 51 (11), pp. 1016–1022.

Kühne, T. 2005. "What Is a Model?". In *Language Engineering for Model-Driven Software Development*, Volume 04101, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI).

Muñoz, P., M. Wimmer, J. Troya, and A. Vallecillo. 2022. "Using Trace Alignments for Measuring the Similarity Between a Physical and its Digital Twin". In *25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 503–510.

Oakes, B. J., B. Meyers, D. Janssens, and H. Vangheluwe. 2021, September. "Structuring and Accessing Knowledge for Historical and Streaming Digital Twins". In *1st Workshop on Ontology-Driven Conceptual Modeling of Digital Twins*, pp. 1–13.

Oakes, B. J., A. Parsai, B. Meyers, I. David, S. Van Mierlo, S. Demeyer, J. Denil, P. De Meulenaere, and H. Vangheluwe. 2023. "A Digital Twin Description Framework and its Mapping to Asset Administration Shell". In *Best papers from MODELSWARD 2021/2022, Springer*. To appear. Available as an arXiv preprint arXiv:2209.12661 [cs.OH].

Oakes, B. J., A. Parsai, S. Van Mierlo, S. Demeyer, J. Denil, P. De Meulenaere, and H. Vangheluwe. 2021. "Improving Digital Twin Experience Reports". In *9th International Conference on Model-Driven Engineering and Software Development - Volume 1: MODELSWARD,*, pp. 179–190. INSTICC, SciTePress.

Spiegel, M., P. Reynolds, and D. Brogan. 2005. "A Case Study of Model Context for Simulation Composability and Reusability". In *Proceedings of the Winter Simulation Conference, 2005.*, Volume 2005, pp. 437–444, Institute of Electrical and Electronics Engineers, Inc.

Stachowiak, H. 1973. *Allgemeine Modelltheorie*. Wien and New York, Springer-Verlag.

Van Acker, B., J. Mertens, P. De Meulenaere, and J. Denil. 2021. "Validity Frame Supported Digital Twin Design of Complex Cyber-Physical Systems". In *2021 Annual Modeling and Simulation Conference (ANNSIM)*, pp. 1–12. Institute of Electrical and Electronics Engineers, Inc.

Van Mierlo, S., B. J. Oakes, B. Van Acker, R. Eslampanah, J. Denil, and H. Vangheluwe. 2020. "Exploring Validity Frames in Practice". In *International Conference on Systems Modelling and Management (ICSMM)*, pp. 131–148, Springer, Cham.

Wright, T., C. Gomes, and J. Woodcock. 2022. "Formally Verified Self-Adaptation of an Incubator Digital Twin". In *International Symposium on Leveraging Applications of Formal Methods*, pp. 89–109. Springer.

Zeigler, B. P. 1976. *Theory of Modelling and Simulation*. New York, Wiley.

AUTHOR BIOGRAPHIES

BENTLEY JAMES OAKES is a post-doctoral researcher at the Université de Montréal. His email is bentley.oakes@umontreal.ca.

CLÁUDIO GOMES is an assistant professor at Aarhus University. His email is claudio.gomes@ece.au.dk.

JOACHIM DENIL is a professor at the Faculty of Applied Engineering at the University of Antwerp. His email is joachim.denil@uantwerpen.be.

JULIEN DEANTONI is a professor at the Université Côte d'Azur - Sophia Antipolis. His email is julien.deantoni@etu.univ-cotedazur.fr.

JOÃO CAMBEIRO is a doctoral student at the Université Côte d'Azur - Sophia Antipolis and at the NOVA University of Lisbon. His email is joao.cambeiro@etu.univ-cotedazur.fr.

JOHN FITZGERALD is a professor and Dean at Newcastle University. His email is john.fitzgerald@ncl.ac.uk.

PETER GORM LARSEN is a professor at Aarhus University. His email is pgl@ece.au.dk.