Examining Model Qualities and Their Impact on Digital Twins

Bentley James Oakes, Cláudio Gomes, Joachim Denil,
Julien Deantoni, João Cambeiro, John Fitzgerald, and Peter Gorm Larsen

Université de Montréal, Aarhus University, Flanders Make and University of Antwerp,
Université Côte d'Azur - Sophia Antipolis, NOVA University, Newcastle University

May 24^{th}, 2023

Motivation

Model Qualities

Impact on DT Services

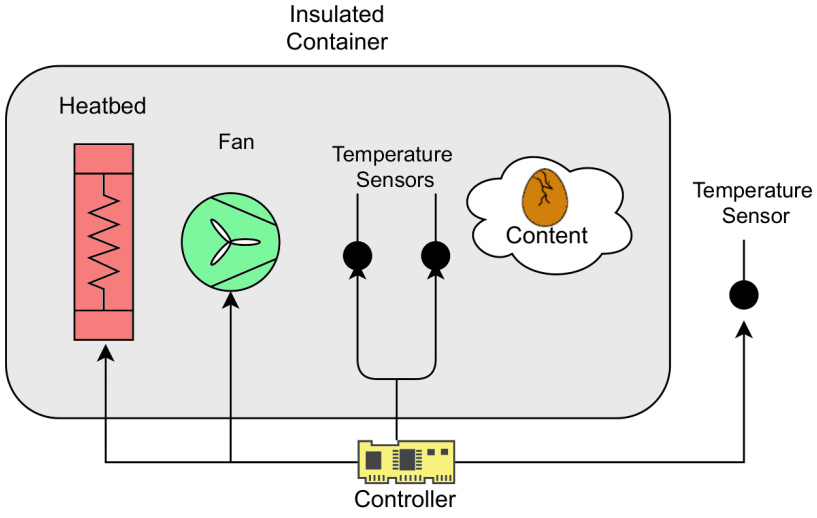
Motivation

Model Qualities

Impact on DT Services

Incubator Running Example

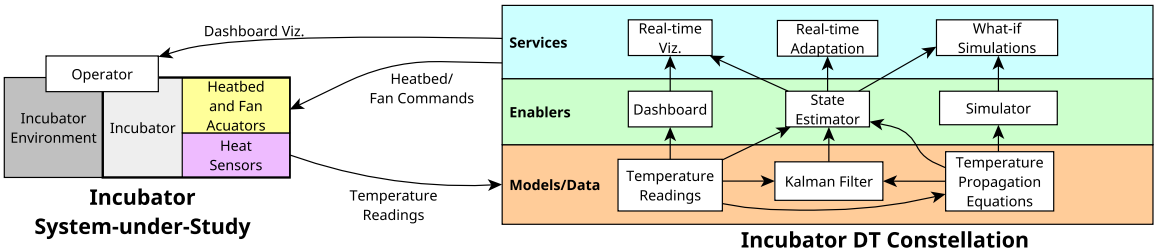
Purpose:
Stable temperature
Egg incubation



Feng *et al.* (2021). *The incubator case study for digital twin engineering*. arXiv preprint arXiv:2102.10390.

Incubator Digital Twin

DTs are a *virtual replica* of a physical system
Offer *services*, to provide *information* or *control*
in *closed-loop* with system



Oakes *et al.* (2022). *A Digital Twin Description Framework and its Mapping to Asset Administration Shell*. arXiv preprint arXiv:2209.12661.

Challenges and Contributions

Challenges:

- DT services relies on models to capture system behaviour
- DT and physical system evolve over time (*degrade*, *upgrade*)

Objective: Define concepts to *discuss quality of DT services*
Go further than “high-fidelity”

Contributions:

- Formally define four *model qualities*
- Discuss impact on DT services
  - Quality degradation makes services *unreliable*

Motivation

Model Qualities

Impact on DT Services

Term	Example	Details
System	Incubator	Box, object, heating element, controller
Environment	Room around incubator	Temperature, table, . . .
Context	Room $\rightarrow$ incubator influence	Ambient room temp. affects heating rate

Properties

1. Warm up from 20 to 30°C in 10 seconds
2. Remain $<$ 80°C at all times

Created for *purpose*, for certain *properties*

Example: Differential equations for *modelling heat propagation*

$$\dot{T}_{heater} = \frac{1}{C_{heater}} \cdot \left(V \cdot I \cdot \Delta t - G_{heater} \cdot (T_{heater} - T_{boxair})\right)$$

$$\dot{T}_{boxair} = \frac{1}{C_{air}} \left(G_{heater} \cdot (T_{heater} - T_{boxair}) - G_{box} \cdot (T_{boxair} - T_{room})\right)$$

Model created by setting parameters

Property Satisfaction

Satisfaction
In particular context, does system satisfy property?
$[\![S]\!]_{C_S} \vDash p$
"In cold room, does incubator reach $80°C$*?"*

Model Satisfaction
In model's context, does model satisfy property?
$[\![M]\!]_{C_M} \vDash p$
"When simulated, do equations reach $80°C$*?"*

Model Qualities:


Relevancy: Does model's context match the system's context?


Verifiability: Can we check property satisfaction on model?


Substitutability: Does satisfaction result match the system result?


Fidelity: How closely do the results match?

Model Quality Consequences

Consider safety property *"does the incubator reach* $80°C$*?"*

Non-relevant model **could** produce **incorrect answer**

Non-verifiable model **can't answer** this question

Non-substitutable model provides **wrong answer**

Non-faithful model provides **erroneous answer**

Model without these qualities is **invalid**

Later: DT services using **invalid** models are **invalid**/**useless**

Relevancy

Model created for particular context

Relevant *only within that context*

Outside context, property satisfaction *may* be unreliable

Example: Temperature propagation equations

Relevant In	**Not Relevant In**
Summer	Winter (below freezing)
Normal room	Mountain-top (low-pressure), space (no air)

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

Capture relevancy with *validity frames* - Denil *et al.*

Can we **verify** whether a model satisfies a property?

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

Example:
Can temperature equations model **provide answer for** "is temperature always $< 80°$C"?

Depends on *model formalism*, *checking methods*, computational resources

- Simulating differential equations can provide an answer
- Formal model-checking may require *more-appropriate* formalism

Can the model **substitute for** the system for property satisfaction?

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

Example:
If temperature model reports that the temperature is always $< 80°C$,
is this true for the system?

How **faithful** is the model to the system?

Example:
How closely do temperature equations match the system's actual temperature?

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

Note: Distance function g **per property**, and error ϵ **per property**

Measure fidelity: Muñoz *et al.*, Biglari and Denil

Model Quality Recap

Relevancy: Does model's context match the system's context?

Verifiability: Can we check property satisfaction on model?

Substitutability: Does satisfaction result match the system result?

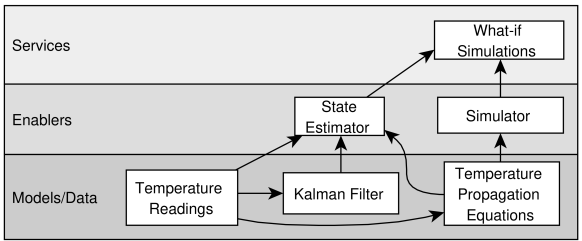
Fidelity: How closely do the results match?

Motivation

Model Qualities

Impact on DT Services

DT Services



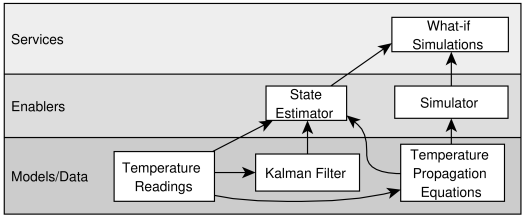
Incubator DT Slice

- Data flows “up” through the DT
- Services rely on enablers
- Enablers rely on models
- *Assumption:* Computation is in the enablers, services just report

Oakes *et al.* (2022). *A Digital Twin Description Framework and its Mapping to Asset Administration Shell*. arXiv preprint arXiv:2209.12661.

Quality Consequences for DT Services

Property: Is incubator always $< 80°C$?



Incubator DT Slice

Relevancy: Relevant service $\leftarrow$ all models relevant

Verifiability: Property must be verifiable for ≥ 1 model

Substitutability: Property must be satisfied by ≥ 1 model

Fidelity: Faithful service $\leftarrow$ all models faithful

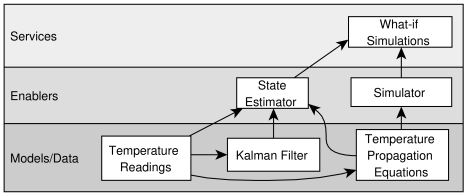
Quality Priority

If the DT or system evolves, order for checking qualities?
Limited time for *safety property*: "temperature $< 80°$C"

1.	**Verifiability**	Model appropriate for property?
2.	**Substitutability**	Satisfaction result? Approximation first?
3.	**Relevancy**	Non-relevant models could be useful
4.	**Fidelity**	Low fidelity results could be useful

Adaptations: Model *switching* or *re-calibration*

Conclusion



Incubator DT Slice

Relevancy: Does DT service's context match the system's context?

Verifiability: Can we check property satisfaction on service?

Substitutability: Does satisfaction result match the system result?

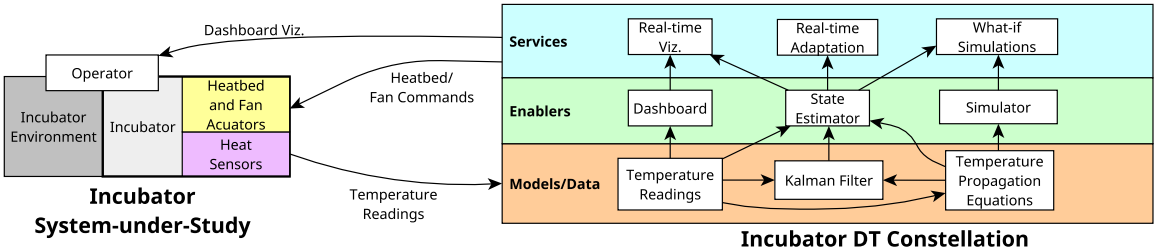
Fidelity: How closely do the results match?

Future work: Integrate into DT framework(s), deepen formalization

Examining Model Qualities and Their Impact on Digital Twins
Oakes, Gomes, Denil, Deantoni, Cambeiro, Fitzgerald, and Larsen
`bentleyoakes.com`

Incubator Digital Twin

DTs are a *virtual replica* of a physical system
Offer *services*, to provide *information* or *control*
in *closed-loop* with system



Oakes *et al.* (2022). *A Digital Twin Description Framework and its Mapping to Asset Administration Shell*. arXiv preprint arXiv:2209.12661.

References

All icons from www.flaticon.com

- Gears, shield, question marks, puzzle piece, and magnifying glass icons made by Freepik
- Down arrow by Kiranshastry
- Lightbulb by Good Ware
- Exchange by itim2101