Towards a Taxonomy of Digital Twin Evolution for Technical Sustainability

Istvan David[*], Dominik Bork[†]
[*]McMaster University, Canada – davidi3@mcmaster.ca
[†]TU Wien, Austria – dominik.bork@tuwien.ac.at

***Abstract*—The next generation of engineered systems ought to be more sustainable. In this context, Digital Twins play a crucial role as key enablers of sustainability ambitions in systems engineering. However, as a specific class of engineered systems, Digital Twins themselves must adopt sustainability principles to avoid defeating their purpose in fostering sustainability. In this proposal, we focus on the technical sustainability of Digital Twins, enabled by their evolution. We propose an initial taxonomy we believe will support systematic Digital Twin evolution mechanisms and draw links to similar taxonomies of Physical Twins.**

***Index Terms*—circular economy, circular systems engineering, model-driven engineering, sustainability, value retention**

I. INTRODUCTION

Sustainability is becoming a key characteristic of modern systems. For example, INCOSE, the International Council on Systems Engineering identifies sustainability as the number one "global megatrend" in systems and their engineering for the decades ahead of us [1]. This trend is driven both by end-users who increasingly voice their preference for human-centered, ecological, economically viable solutions; and by governments that realize the unsustainability of the status quo.[1]

Digital Twins are live virtual representations of physical systems with control capabilities over the said physical system [2]. By these traits, Digital Twins enable complex reasoning and automated control of the physical system, allowing for better resource management, better waste prediction, and the collection of important data throughout the lifetime of the physical twin. As shown in Fig. 1, these capabilities position Digital Twins particularly well in supporting sustainability in complex systems—from smart manufacturing [3] to precision agriculture [4]—evidenced by the rapidly growing related body of knowledge on sustainability by [5] and of [6] Digital Twins, and novel systems engineering paradigms [7].

However, as a specific class of engineered systems, Digital Twins themselves must become sustainable (Fig. 1). Unsustainable Digital Twins defeat the purpose of sustainability efforts by moving sustainability issues from the engineered system into the engineering methodology [8]. Clearly, we must strive to build sustainable Digital Twins [9] if we want to build sustainable systems by them. Due to the complex nature of sustainability, however, this is a challenging task.

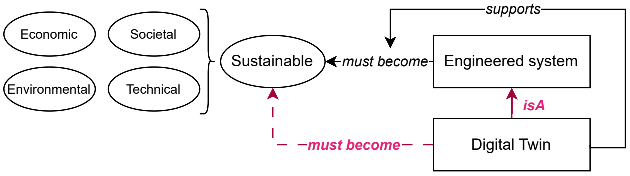

Fig. 1. The relationship between systems, Digital Twins, and sustainability

Augmenting Brundtland's classic, three-dimensional model of sustainability [10], consisting of *economic* (financial viability), *environmental* (reduced impact), and *social* (elevated utility) sustainability dimensions, Lago et al. [11] define a fourth dimension, *technical sustainability* as the ability of a system to be used over an extended lifetime (Fig. 1). The core enablers of technical sustainability are appropriate evolution mechanisms [11] that help adapt the system to a changing environment, requirements, and goals of the system. While self-adaptation and evolution of Digital Twins have received some attention lately [12], [13], no comprehensive collection of elementary Digital Twin evolution mechanisms exists. This substantially hinders the development of systematic approaches to Digital Twin evolution, and by extension, limiting the technical sustainability of Digital Twins.

To address the limitations of the state-of-the-art, in this paper, we define an initial taxonomy for Digital Twin evolution in support of their technical sustainability. We draw from our hands-on experiences in developing Digital Twins and follow established methods for taxonomy building [14], [15]. Our goal is to initiate research and call to action in support of sustainable systems engineering by sustainable Digital Twins. The modeling community is particularly well-positioned to lead coherent sustainability efforts related to complex systems due to its broader systems view and specialized sub-communities.

This paper is the first step towards a coherent and actionable taxonomy to aid systems engineers in implementing sustainable Digital Twins and assessing their sustainability properties.

Our taxonomy defines seven evolution mechanisms in the form of *R-imperatives*, a concept widely applied in domains of physical systems where sustainability has become an urgent need in recent years, e.g., reverse logistics, waste management, and circular economy [16]. By this choice, we wish to draw the attention of the modeling community to the vast body of knowledge on value retention mechanisms available for adoption on the digital side of Digital Twins.

---

[1] For example, see the Industry 5.0 initiative of the European Commission: https://research-and-innovation.ec.europa.eu/research-area/industrial-research-and-innovation/industry-50_en.

II. BACKGROUND AND RELATED WORK

Here, we review some key concepts and related works.

A. *R-imperatives and frameworks*

R-imperatives specify mechanisms for value retention. With a varying number of R-imperatives, frameworks such as 3R, 6R, and 10R are well-established in domains where environmental and ecological impacts are obvious, such as waste management and closed-loop supply chain management. Perhaps the most known of such frameworks is 3R advocating for reduce, reuse, and recycle—that is, minimize waste, use items more than once, and find new use instead of throw things away, respectively. Different domains adopt their own finer or coarser-grained R-imperatives with varying rigor and details. The 10R framework of Reike at al. [16] offers a recent synthesis that comprises ten mechanisms that, depending on the length of the engineering subprocess they span, are classified as short-loop (refuse, reduce, resell/reuse, repair), medium-loop (refurbish, remanufacture, repurpose), and long-loop mechanisms (recycle, recover energy, remine materials).

Having gained popularity in government regulations and business strategies [16], there exists a substantial body of knowledge on R-imperatives in the realm of physical systems. This knowledge could be transposed to the digital realm to support Digital Twin evolution to avoid reinventing the wheel. However, the actionability of R-frameworks is hindered by a handful of factors. Only about half of the known R-frameworks define clear mechanisms and the relationships between them [16]. Furthermore, defining sustainability objectives and using R-imperatives to achieve those objectives in systems engineering is challenged by the often overlapping concerns of different R-imperatives [17].

To tackle these limitations of R-frameworks, our taxonomy clearly relates R-imperatives to specific artifacts of a Digital Twin context (Fig. 2 and Table I), and defines clear, causal relationships among them (Fig. 3).

B. *Sustainability by and of Digital Twins*

Sustainability of Digital Twins is becoming a research topic of particular interest. The majority of the state of the art focuses on sustainability *by* Digital Twins, but recently, works on the sustainability *of* Digital Twins have appeared as well.

Considering the former, Digital Twins are well-positioned to govern end-to-end processes [9], contributing to important operational goals, such as adaptive control and predictive maintenance. However, Digital Twins are no silver bullet either, as reported by Tzachor et al. [18], due to the multitude of challenges that might prevent Digital Twins to support sustainability goals, e.g., social divides and economic inequalities.

Considering the latter, Bellis and Denil [6] report four sustainability challenges of Digital Twinning: energy consumption, modeling effort and complexity, the ability to evolve with the physical twin, and the deployment of the twin architecture within organizations. Our work puts the latter two concerns in context with a focus on the technical dimension of sustainability. Fur et al. [5] investigate the three classic dimensions of sustainability by Brundtland, but technical sustainability is out of their scope. Our taxonomy complements their work.

While all these works are valuable in targeting specific sustainability challenges, they do not provide a holistic and structured overview of sustainability mechanisms. This is understandable, as that would require a *taxonomy*. Our work provides just that: an initial taxonomy—however, it only focuses on the technical dimension of sustainability.

C. *Taxonomies*

A taxonomy is a form of classification [14], aiming to systematically organize knowledge of a specific research field or problem. Classification of objects helps to understand the specific field and to build theories [19], [20]. The terms taxonomy, typology, and framework, are sometimes used interchangeably [14]. Some of the influential taxonomies in software and systems modeling related to our concerns are the taxonomy of software change by Buckley et al. [21] and the model transformations taxonomy of Mens and Van Gorp [22].

Following the taxonomy-building process by Nickerson et al. [14] and the case-based generalization approach of Wierenga and Daneva [15], we draw from our hands-on experiences in developing Digital Twins to define our taxonomy. First, we analyzed repeatedly emerging phenomena in our previous projects. Second, we decomposed these phenomena architecturally. Third, we generalized the phenomena to architecturally similar cases. Finally, we organized evidence by minimal, compact taxonomy.

III. A 7R TAXONOMY OF DIGITAL TWIN EVOLUTION

Our taxonomy comprises seven R-imperatives, shown in Fig. 2 in the context of a general digital twinning scenario.

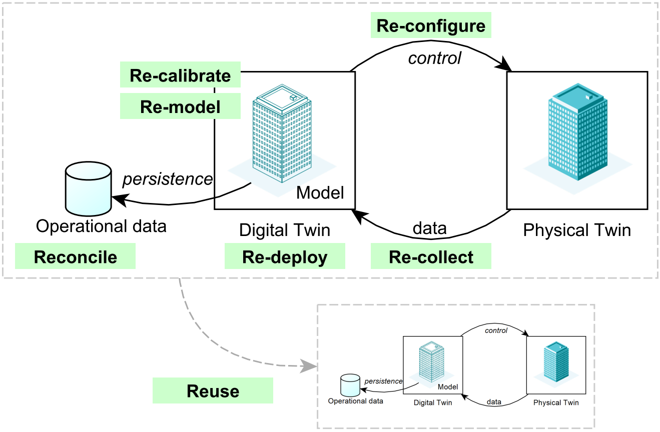

Fig. 2. R-imperatives of the taxonomy in a Digital Twin context

The R-imperatives are further classified into four different realms characteristic of Digital Twins (Table I). These realms are also found in relevant reference frameworks, such as the one defined by the ISO 23247 standard [23].

Two R-imperatives are concerned with the *Model* at the core of the *Digital Twin* (*re-calibrate, re-model*). Two R-imperatives are concerned with the *data*, specifically, how it

TABLE I
R-IMPERATIVES, THEIR REALMS, AND TARGET ARTIFACTS

R-imperative	**Realm**	**Target**
Re-calibrate	Model	Parameters
Re-model	Model	(Meta)models
Re-collect	Data	Operational data
Reconcile	Data	Database
Re-deploy	System	Digital Twin
Re-configure	System	Physical Twin
Reuse	System & Process	Digital Thread

More complex ↓

More frequent ↑

is stored (*reconcile*) and how is it obtained from the *Physical Twin* (*re-collect*). Two R-imperatives are concerned with the twin system as a whole (*re-deploy, re-configure*). Finally, one R-imperative is concerned with translating the know-how accumulated during the lifetime of the Digital Twin into other digital twinning scenarios (*reuse*).

As indicated in Table I, the complexity of R-imperatives increases with the complexity of the target artifact. For example, re-calibrating a parameter is substantially less elaborate than reconciling a database. This is due to some R-imperatives being dependent on each other and necessitating the execution of others. For example, upon reconciling a database, models might have to be updated as well, further necessitating re-calibration of crucial parameters. Such dependencies are often referred to as short-loop and long-loop sustainability imperatives in physical systems [16]. Understanding such dependencies is key in utilizing the taxonomy. Fig. 3 shows a possible map of the system's state space, with the R-imperatives and their causes linked to each other.

In addition, as shown in Fig. 3, maintaining a *correct system* requires *correct models* and *correct data*. This means R-imperatives of the system realm can only be executed when all models and data are correct.

We now elaborate on the seven R-imperatives, using Fig. 3.

A. Re-calibrate

Re-calibration of a model's parameter is required when parameter drift occurs and the model is not a faithful representation of the physical twin anymore. This discrepancy gives rise to incorrect simulations and a subsequent imprecise control of the physical twin. Re-calibration is firmly situated at the run-time phase of the system. Often, manual re-calibration is not acceptable and automated means are required. For example, reinforcement learning has been used for this purpose [24]. As the most primitive R-imperative, re-calibration does not depend on any other R-imperatives; however, other R-imperatives might require re-calibration, as shown in Fig. 3.

B. Re-model

In more elaborate cases concept drift may occur, i.e., the model does not reflect the real phenomenon properly. For example, new components have been added to the physical system or a newly identified phenomenon must be encompassed in the model. Such cases might require substantial changes to the model beyond re-calibrating parameters. Specific modeling and software engineering tasks might be considered as refinements of this R-imperative, including repair of models [25], re-architecting a Digital Twin, or re-packaging a software component to match a changing API. After re-modeling, models need to be re-calibrated, imposing a dependency on the previously discussed R-imperative. This dependency is shown in Fig. 3 with an arrow from re-modeling to re-calibration. Often, automated model adaptation is required, e.g., by computer-aided inference and adaptation of Digital Twin simulation models [26].

C. Reconcile

Data is strongly related to models [27]. During the design phase, data is used for building models, and during operation, data is analyzed through the view of specific models, augmenting data with additional semantics, resulting in information. If data discrepancies occur, data might become inconsistent. Such cases might be caused by software or hardware issues, e.g., an unhandled exception in the software code of the Digital Twin, or a failing sensor in the Physical Twin. In these cases, data needs to be reconciled, i.e., its schema must be updated and data must be properly migrated. As shown in Fig. 3, reconciliation might require re-modeling the system accordingly. Conversely, as shown by the red arrow that points from Re-modeling to Data discrepancy in Fig. 3, data discrepancy might be caused by a changing model that serves as the data schema. That is, re-modeling might necessitate reconciliation. Such circular dependencies are especially problematic, but understanding that such co-dependencies exist alleviates the complexity of putting the right procedures in place.

D. Re-collect

Transient event blackouts—i.e., grayouts—result in missing data on the sensor event stream. Such cases might occur due to transient failures, or in situations when the Digital Twin must evolve but functionality cannot be hot-swapped, e.g., during reconciling the database. It is paramount to be able to re-collect such data. Decoupling data collection from data processing and persistence is a typical architectural choice to support such a mechanism. After re-collecting data, reconciliation, re-modeling, and re-calibration might be needed.

E. Re-deploy

After ensuring that models and data are correct, system-level correctness must be managed. In cases when the models or database of the Digital Twin undergo evolution, the Digital Twin version in use is an outdated one. Thus, the evolved version of the Digital Twin must be rolled out to be used in production, i.e., the Digital Twin needs to be re-deployed [28].

F. Re-configure

The ultimate goal of the Digital Twin is to control the Physical Twin in an optimal fashion. As the Physical Twin evolves, it might not operate under the optimal circumstances,

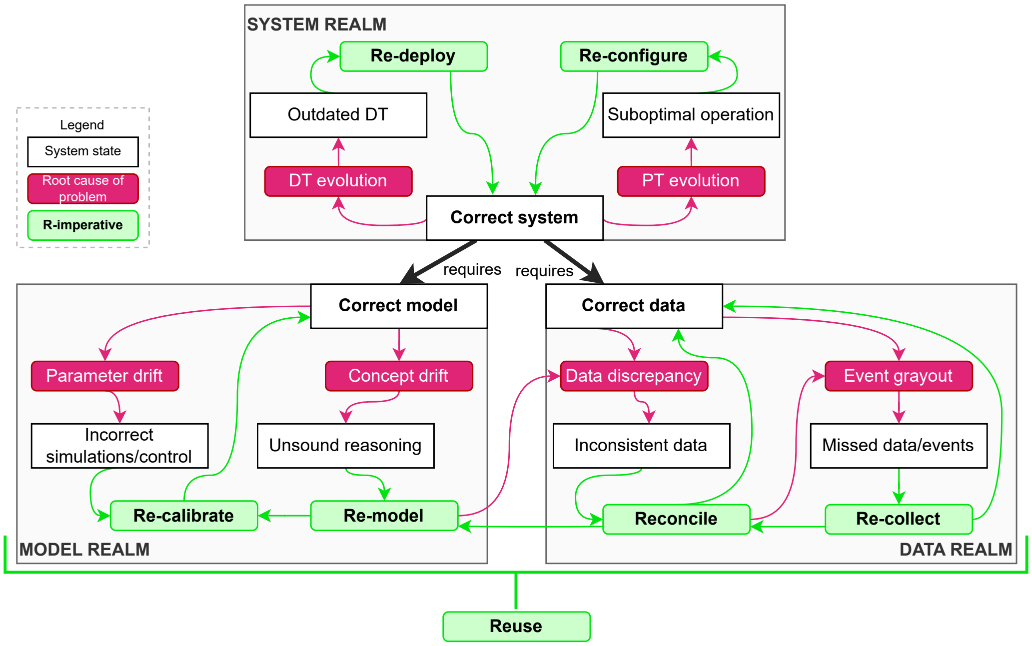

Fig. 3. Possible operationalization of the R-imperatives

and therefore, it needs to be re-configured by the Digital Twin. This mechanism is well-understood in Digital Twins. The current taxonomy merely positions it within a larger context.

G. Reuse

During the maintenance and evolution of the Digital Twin, large amounts of data, knowledge, and know-how are recorded. To retain the value of these artifacts, they should be reused in other digital twinning situations. Aptly, it is often Digital Twins that are used as the authoritative source of data in modern engineering processes. Thanks to the advanced data collection mechanisms of Digital Twins, data availability is not a bottleneck anymore in reuse. The main challenge is enriching data with appropriate semantic information to create intelligence [27]. Automation by transfer learning is becoming an increasingly viable option in the era of big data [29].

IV. RECOMMENDED RESEARCH DIRECTIONS

We now outline some research directions for prospective scholars we anticipate to be impactful.

A. Evaluation and gradual extension

As with any taxonomy, it is important to put it to use to test its capabilities. Our work was focusing on the problem identification and taxonomy development phases of the standard taxonomy development process by Kundisch et al. [30]. This leaves space for prospective researchers and practitioners to work on the demonstration and evaluation of the taxonomy. We recommend contributing exemplars and case studies to this end. To extend and enrich the taxonomy, we recommend studying how the major influencing factors of technical sustainability of software [31] translate to Digital Twin settings. Quality requirements, such as interoperability and availability might give rise to additional R-imperatives.

B. Mapping onto lifecycle models and architectural models

The operationalization of the taxonomy in Fig. 3 spans a process. Mapping the taxonomy onto lifecycle models will provide more actionable directives to apply the taxonomy in real systems engineering settings. Alignment with DevOps [32] and TwinOps [33] are particularly promising research directions. Similarly, mapping the taxonomy onto architectural models will allow for architectural choices that account for the systematic and sustainable evolution of Digital Twins. We particularly encourage investigating alignment with the ISO 23247 standard of Digital Twins for manufacturing [23]. However, emerging alternative architectural choices for Digital Twins deserve attention as well [34], [12].

C. Link with R-frameworks of physical systems

The current work deliberately focused on Digital Twins, but R-imperatives as value retention mechanisms are well-researched in systems that are subject to becoming Physical Twins [16]. Naturally, establishing a link with existing R-frameworks of physical systems will allow for a more holistic view of evolutionary mechanisms, eventually allowing for digital-physical twin co-evolution. We advocate for an interdisciplinary approach to effectively tackle the highly multi-systemic nature of sustainability [35].

D. Tools and automation

To leverage the taxonomy to its fullest extent, we recommend investigating possible means of tool support and the automation of evolution processes. Especially in long-loop evolution mechanisms, such as data re-collection and reconciliation, automation will provide substantial benefits. Analysis and simulation tools for planned evolution, and build tools that can integrate into DevOps processes are particularly important directions. A list of contemporary tools is provided by Muctadir et al. [36].

V. CONCLUSION

In this paper, we presented an initial taxonomy for Digital Twin evolution. System evolution is a key mechanism in extending the useful lifetime of systems, i.e., fostering technical sustainability. As sustainability is becoming a leading principle in the next generation of systems, the technical dimension is the most tangible for those who develop systems. Improving the sustainability in Digital Twins is particularly important as this class of systems is considered as an important aid in fostering sustainability in large-scale complex systems.

Our taxonomy is meant to help identify specific actions Digital Twin frameworks can implement in support of evolution. Our taxonomy might also help researchers in identifying opportunities for their research efforts, especially in modeling sub-communities such as the ones focusing on models and evolution, multi-paradigm modeling of CPS, and DevOps.

This paper aims to raise awareness of the body of knowledge on sustainability in various domains of engineered systems from which important know-how could be adopted by the modeling community and transposed to digital systems.

In future work, we plan to apply the framework in various case studies to test its completeness and preferably, encounter new R-imperatives in support of Digital Twin evolution and Digital-Physical Twin co-evolution.

REFERENCES

[1] INCOSE, “Systems Engineering Vision 2035,” https://www.incose.org/about-systems-engineering/se-vision-2035, 2022.

[2] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, “Digital twin in manufacturing: A categorical literature review and classification,” *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.

[3] J. Moyne *et al.*, “A requirements driven digital twin framework: Specification and opportunities,” *IEEE Access*, pp. 107 781–107 801, 2020.

[4] I. David *et al.*, “Digital Twins for Cyber-Biophysical Systems: Challenges and Lessons Learned,” in *ACM/IEEE 26th Intl. Conf. on Model-Driven Engineering Languages and Systems (MODELS)*. IEEE, 2023.

[5] S. Fur, M. Heithoff, J. Michael, L. Netz, J. Pfeiffer, B. Rumpe, and A. Wortmann, *Sustainable Digital Twin Engineering for the Internet of Production*. Springer, 2023, pp. 101–121.

[6] S. Bellis and J. Denil, “Challenges and possible approaches for sustainable digital twinning,” in *Proceedings of the 25th Intl. Conf. on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS 2022, Montreal, Canada, 2022*. ACM, 2022, pp. 643–648.

[7] I. David, D. Bork, and G. Kappel, “Circular Systems Engineering,” Tech. Rep., 2023, https://arxiv.org/abs/2306.17808.

[8] P. Lago, *The Digital Society Is Already Here – Pity It Is 'Unsustainable'*. VU University Press Amsterdam, Jun. 2023.

[9] M. Heithoff, A. Hellwig, J. Michael, and B. Rumpe, “Digital twins for sustainable software systems,” 2023.

[10] G. H. Brundtland, “Our common future–call for action,” *Environmental conservation*, vol. 14, no. 4, pp. 291–294, 1987.

[11] P. Lago, S. A. Koçak, I. Crnkovic, and B. Penzenstadler, “Framing sustainability as a property of software quality,” *Commun. ACM*, vol. 58, no. 10, pp. 70–78, 2015.

[12] P. Bibow *et al.*, “Model-driven development of a digital twin for injection molding,” in *Advanced Information Systems Engineering - 32nd International Conference, CAiSE 2020, Grenoble, France, 2020, Proceedings*, ser. LNCS, vol. 12127. Springer, 2020, pp. 85–100.

[13] J. A. Erkoyuncu, I. F. del Amo, D. Ariansyah, D. Bulka, R. Vrabič, and R. Roy, “A design framework for adaptive digital twins,” *CIRP Annals*, vol. 69, no. 1, pp. 145–148, 2020.

[14] R. C. Nickerson, U. Varshney, and J. Muntermann, “A method for taxonomy development and its application in information systems,” *European Journal of Inf. Systems*, vol. 22, no. 3, pp. 336–359, 2013.

[15] R. Wieringa and M. Daneva, “Six strategies for generalizing software engineering theories,” *Science of Computer Programming*, vol. 101, pp. 136–152, 2015.

[16] D. Reike, others Walter J.V. Vermeulen, and S. Witjes, “The circular economy: New or Refurbished as CE 3.0? – Exploring Controversies in the Conceptualization of the Circular Economy through a Focus on History and Resource Value Retention Options,” *Resources, Conservation and Recycling*, vol. 135, pp. 246–264, 2018.

[17] W. M. P. van der Aalst, O. Hinz, and C. Weinhardt, “Sustainable systems engineering,” *Business & Information Systems Engineering*, vol. 65, no. 1, pp. 1–6, 2023.

[18] A. Tzachor *et al.*, “Potential and limitations of digital twins to achieve the sustainable development goals,” *Nature Sustainability*, vol. 5, no. 10, pp. 822–829, 2022.

[19] D. H. Doty and W. H. Glick, “Typologies as a unique form of theory building: Toward improved understanding and modeling,” *Academy of Management Review*, vol. 19, no. 2, pp. 230–251, 1994.

[20] M. Tory and T. Möller, “Rethinking visualization: A high-level taxonomy,” in *10th IEEE Symposium on Information Visualization (InfoVis 2004)*, M. O. Ward and T. Munzner, Eds. IEEE, 2004, pp. 151–158.

[21] J. Buckley, T. Mens, M. Zenger, A. Rashid, and G. Kniesel, “Towards a taxonomy of software change,” *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 17, no. 5, pp. 309–332, 2005.

[22] T. Mens and P. Van Gorp, “A taxonomy of model transformation,” *Electronic Notes in Theoretical Comp. Sci.*, vol. 152, pp. 125–142, 2006.

[23] G. Shao, “Use case scenarios for digital twin implementation based on ISO 23247,” Tech. Rep., 2021, https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.400-2.pdf.

[24] N. Tomin *et al.*, “Development of digital twin for load center on the example of distribution network of an urban district,” in *E3S Web of Conferences*, vol. 209, 2020.

[25] P. Stünkel, H. König, A. Rutle, and Y. Lamo, “Multi-model evolution through model repair,” *Journal of Object Technology*, vol. 20, no. 1, pp. 1:1–25, Jan. 2021, workshop on Models and Evolution (ME 2020).

[26] I. David and E. Syriani, “DEVS Model Construction as a Reinforcement Learning Problem,” in *2022 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, 2022, pp. 30–41.

[27] B. Combemale *et al.*, “A hitchhiker's guide to model-driven engineering for data-centric systems,” *IEEE Software*, vol. 38, no. 4, pp. 71–84, 2021.

[28] A. Göppert, L. Grahn, J. Rachner, D. Grunert, S. Hort, and R. H. Schmitt, “Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems,” *J. of Intelligent Manufacturing*, vol. 34, no. 5, pp. 2133–2152, 2023.

[29] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big Data*, vol. 3, no. 1, p. 9, May 2016.

[30] D. Kundisch, J. Muntermann, A. M. Oberländer, D. Rau, M. Röglinger, T. Schoormann, and D. Szopinski, “An update for taxonomy designers,” *Business & Inf. Systems Engineering*, vol. 64, no. 4, pp. 421–439, 2022.

[31] N. Condori-Fernandez and P. Lago, “Characterizing the contribution of quality requirements to software sustainability,” *Journal of Systems and Software*, vol. 137, pp. 289–305, 2018.

[32] S. Mosser and J.-M. Bruel, “Requirements engineering in the devops era,” in *2021 IEEE 29th International Requirements Engineering Conference (RE)*, 2021, pp. 510–511.

[33] J. Hugues, A. Hristosov, J. J. Hudak, and J. Yankel, “Twinops – devops meets model-based engineering and digital twins for the engineering of cps,” in *Proc. ACM/IEEE Intl. Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. ACM, 2020.

[34] E. Ferko, A. Bucaioni, M. Behnam, and P. Pelliccione, “Analysing interoperability in digital twin software architectures for manufacturing,” in *17th European Conference on Software Architecture*, 2023.

[35] S. McGuire, E. Shultz, B. Ayoola, and P. Ralph, “Sustainability is stratified: Toward a better theory of sustainable software engineering,” *CoRR*, vol. abs/2301.11129, 2023.

[36] H. M. Muctadir, D. A. M. Negrin, R. Gunasekaran, L. Cleophas, M. van den Brand, and B. R. Haverkort, “Current trends in digital twin development, maintenance, and operation: An interview study,” *CoRR*, vol. abs/2306.10085, 2023.