Digital Twin System of Systems: A Layered Architecture Proposal

Meriem Smati[1,2][a], Vincent Cheutet[1][b], Christophe Danjou[2][c] and Jannik Laval[1][d]

[1] *INSA Lyon, Université Lumière Lyon 2, Université Claude Bernard Lyon 1, Université Jean Monnet Saint-Etienne, DISP UR4570, Villeurbanne, 69621, France*

[2] *Laboratoire Poly-Industrie 4.0, Département de Mathématiques Et Génie Industriel, Polytechnique Montréal, Montréal, Québec, Canada*

{*meriem.smati, vincent.cheutet*}*@insa-lyon.fr, christophe.danjou@polymtl.ca, jannik.laval@univ-lyon2.fr*

Keywords: Digital Twins, System of Systems, Model-Driven Architecture (MDA), Model-Driven Engineering (MDE), Dynamic Modeling.

Abstract: Integrating Digital Twins (DTs) with Systems of Systems (SoSs) offers transformative potential for optimizing complex, interconnected systems. However, implementing a DT for an SoS poses several challenges due to the independence and diversity of Constituent Systems (CSs), as well as SoS-specific characteristics such as geographic distribution, evolutionary development, and emergent behavior. This study proposes a novel architectural framework for an SoS DT, featuring a layered design that combines individual DTs for each CS with a global SoS DT layer to oversee and coordinate their interactions. By bridging limitations found in standalone DTs, this structure enables a cohesive and adaptive digital representation of the SoS, addressing the challenges of autonomy and extensibility. The framework aligns with fundamental SoS characteristics, paving the way for enhanced system management, predictive analysis, and performance monitoring, while also underscoring the need for a standardized metamodel to support resilient SoS DT development.

1 INTRODUCTION

A System of Systems (SoS) refers to a form of complex systems wherein multiple, autonomous systems, each with independently defined functions, operations and objectives, cooperate to achieve a bigger common goal (Checkland, 1999). This interaction between Constituents Systems (CSs) not only improves communication and interoperability but also creates new capabilities that go beyond the sum of what each system could achieve on its own (Maier, 1998). SoS designs are widely used in areas like transportation, defense, healthcare, and smart cities, where systems are complex and sometimes critical and need to be flexible, efficient and resilient to keep operations stable in changing situations (DeLaurentis, 2005).

Digital Twins (DTs) are a concept that were first introduced by Michael Grieves in 2003 (Grieves and Vickers, 2017) and since its creation, it has been defined in numerous ways, including:

- "In the context of Industry 4.0, the Digital Twin is introduced as a framework for **mirroring** certain aspects of the underlying physical entities in the manufacturing processes" (Josifovska et al., 2019).
- "A DT consists of a virtual representation of a production system that is able to use sensory data, connected smart devices, mathematical models, and real-time data elaborations. The DT can be run on different simulation disciplines that are characterized by the **synchronization** of virtual and real systems" (Tan et al., 2019).
- "A digital twin is a virtual representation of a physical product or process, used to understand and **predict** their performance." (Rauch and Pietrzyk, 2019).
- "The digital twin is an integrated system with low-cost IoT sensors to gather system data, advanced data analytics to draw meaningful insights and predictive maintenance strategy based on the machine learning algorithm to **reduce preventive maintenance cost**. Overall the digital twin act as a digital replica of the field asset which is monitored and maintained based on actual sensor data from the physical field using machine learning." (Bhowmik, 2019).

[a] https://orcid.org/0009-0006-9399-3356
[b] https://orcid.org/0000-0003-1920-2609
[c] https://orcid.org/0000-0002-9575-0087
[d] https://orcid.org/0000-0002-7155-5762

Smati, M., Cheutet, V., Danjou, C. and Laval, J.
Digital Twin System of Systems: A Layered Architecture Proposal.
DOI: 10.5220/0013258400003896

In *Proceedings of the 13th International Conference on Model-Based Software and Systems Engineering (MODELSWARD 2025)*, pages 273-279
ISBN: 978-989-758-729-0; ISSN: 2184-4348

These abilities of **mirroring, synchronizing, predicting and lowering maintenance cost** made it gain considerable traction and emerging as a crucial element in advances in different applications and domains, such as Industry 4.0, aerospace and manufacturing (Broo et al., 2022) and this explains why DTs have been chosen as the focus of this paper; specifically, our objective is to represent a DT for an SoS that can simulate, analyze, and predict the behavior and performance of the SoS.

Implementing a DT of a SoS is challenging due to the specific criteria and characteristics of SoS, i.e operational and managerial independence of CSs (Maier, 1998), varying levels of autonomy, scalability and real-time data processing capability. Therefore, these hurdles of heterogeneity and independence have lead to propose a main research question (RQ) which is: **RQ: How can a DT be modeled to accurately represent a physical SoS, aiming to enhance resilience?**

We divide this RQ into sub-questions:

- RQ1.1: Should the DT be a single, overarching DT, or a collection of interconnected DTs for each CS?
- RQ1.2: If a multi-DTs approach is adopted, is a *system of DTs* enough to represent the SoS?
- RQ1.3: If a SoS DT cannot be represented by the *system of DTs*, what additional layer is necessary for accurate representation?
- RQ1.4: Is the proposed architecture consistent with the fundamental characteristics of SoS?

The rest of this paper is organized and outlined herein. Section 2 covers the related works on SoS and DT, Section 3 introduces the proposed concept, followed by a detailed description of the SoS DT in Section 4. Section 5 presents theoretical use cases, and finally, Section 6 concludes the paper with a discussion and insights into future work.

2 RELATED WORK

The intersection of DTs and SoS has sparked significant academic and practical interest, leading to a growing body of literature that explores how these two concepts can be integrated to enhance system efficiency, reliability, and functionality. Although many articles touch upon DTs and SoS separately, only a few delve into both concepts within the same study especially for resilience purposes.

Broo et al. (Broo et al., 2022) have examined different existing DTs architectures and proposed a case study of the design and implementation of a DT for a smart infrastructure, in particular a railway bridge. Even though the article highlights the importance of adopting a SoS perspective in DT design, it does not present the creation of a complete SoS DT. However the case study highlights the essential considerations for moving towards DTs capable of representing and interacting with larger systems. Integration and services are key elements of this vision.

On the same perspective of needing a DT of a SoS, Demir et al. (Demir et al., 2023) explains how to create a DT architecture for a SoS by proposing a holistic framework for DTs that spans several hierarchical levels, i.e product level, process level, system level and SoS level. The aim of this architecture was to enable rapid reconfiguration of production lines and dynamic networks to adapt to new production requirements.

Pickering et al. (Pickering et al., 2023) does not give precise instructions on how to create a DT of a SoS. However, it does propose a concept, called Modular Agritech Systems for Horticulture (MAS-H), which could serve as a basis for developing such an architecture in the horticultural field, and more specifically for the kiwifruit industry in New Zealand. This horticultural SoS DT offers a reduction in technology life-cycle costs, an improved collaboration and data sharing, and aid in decision-making.

Azari and al. (Azari et al., 2022) consider a set of interconnected cyber-physical systems (CPS) a SoS and focuses on the use of Transfer Learning (TL) to improve the resilience of the CSs. And the DT in this context is used to host runtime models that reflect the behavior of CSs in the SoS, provide data for training TL-based predictive maintenance models and facilitate data and knowledge sharing.

Parri et al. (Parri et al., 2021) highlights a hardware/software framework called JARVIS, designed to improve the resilience of a CPS, particularly in the contexts of smart cities and the Industrial Internet of Things (IIoT). If we consider a CPS a SoS and although the article does not provide a guide to creating a DT architecture for SoSs, it offers valuable information on how DTs can be used in such a context (such as improving operation, integration, maintenance and recoverability).

Olsson and Axelsson (Olsson and Axelsson, 2023) survey the current state of DTs in SoS, proposing two architectures: one DT for the entire SoS or individual DTs for each component. The monolithic approach faces scalability, single-point failure, and integration issues, while the distributed approach encounters interoperability, synchronization, and coordination challenges.

Borth et al. (Borth et al., 2019) cites that implementing DTs for cyber-physical SoS and IoT installations presents specific challenges due to the dynamic nature of SoS, the operational independence of CSs and data sharing issues. To address these challenges, architectural strategies for SoS DTs can focus on the upper echelons of the information hierarchy, adopt a modular, causality-based approach to structuring internal models, integrate reflection mechanisms for self-assessment of performance, and utilize points of loose coupling within the SoS for data connection between digital and Physical Twins (PTs). The authors mentioned that SoS DTs can help overcome the challenges associated with SoS management, knowledge management, unexpected emergent effects and the additional costs associated with updates and upgrades.

In summary, most articles address domain-specific challenges and contributions related to ”SoS” and ”DT” separately rather than focusing on developing a ”DT of a SoS”. Among those that do attempt it, few succeed in establishing a metamodel or guidelines for creating an SoS DT.

3 INTRODUCING THE SoS DT CONCEPT

In this section, we introduce an on-top architecture designed to address the previously stated RQ, aligning with the characteristics of both the SoS and the DT.

3.1 Adoption of a Multi-DTs Approach for SoSs (RQ1.1)

As mentioned by Maier (Maier, 1998) ”A system-of-systems is an assemblage of components which individually may be regarded as systems”, and ”A system that has operational and managerial independence of its elements is a system-of-systems. But a system composed of complex subsystems that do not have both operational and managerial independence is not a system-of-systems, no matter the complexity of the subsystems”.

Operational independence is the ability of a system to effectively function on its own and deliver valuable services without relying on the larger system. This means that even when separated from the SoS, each component can fulfill its intended purpose. And managerial independence denotes that CSs are not only ”capable” of operating autonomously but ”actively do” so. They are acquired and maintained separately, ensuring that they retain their operational capabilities independently of the overarching SoS (Maier, 1998). And since the DT needs to mirror the physical system which in this case is a SoS, **a multi-DTs approach must be adopted** to match the operational and managerial independence required on the physical SoS since each CS is considered a fully independent system working on its own, and this explanation would justify and answer the sub-question RQ1.1.

For a better explanation, Figure 1.A illustrates two systems, S1 and S2, engaging in communication. This interaction can be either bidirectional or unidirectional based on specific requirements. Each system communicates with its respective DT. Here, the previously mentioned criteria are respected on the digital and physical side but at this stage, there is no SoS present even though operational and managerial independence are respected; rather, there are only two individual systems in communication due to the lack of common objective or objectives.

3.2 The Limitations of a System of DTs in Representing a SoS DT (RQ1.2)

To address RQ1.2, we consider a simple SoS consisting of two CSs, as illustrated in Figure 1.B, each System has its own DT with the bidirectional medium for an effective communication between the physical and digital asset and a common objective is shared between S1 and S2. On the digital side, if this communication is mirrored between DT1 and DT2, a *System of DTs* is established. For improved clarity, we define the *system of DTs* as ”a system that compromises the digital replicas of CSs that interact with each other at a software level to mirror the behavior of the physical communication”.

The *system of DTs* could be sufficient to represent to replicate the SoS, given that it comprises only a few systems and the communication is not complex. For example, the communication between S1 and S2 could simply be represented by a MQTT[1], Kafka[2] or any other type of communication protocol that can replicate the behavior of the physical interaction digitally depending on the data types.

Nevertheless, there are scenarios where a *system of DTs* is inadequate to fully represent the SoS. For example, when a system within the SoS communicates simultaneously with multiple other systems to achieve the shared objectives, several critical questions arise:

- Is the computational power sufficient?

[1]https://mqtt.org/

[2]https://kafka.apache.org/

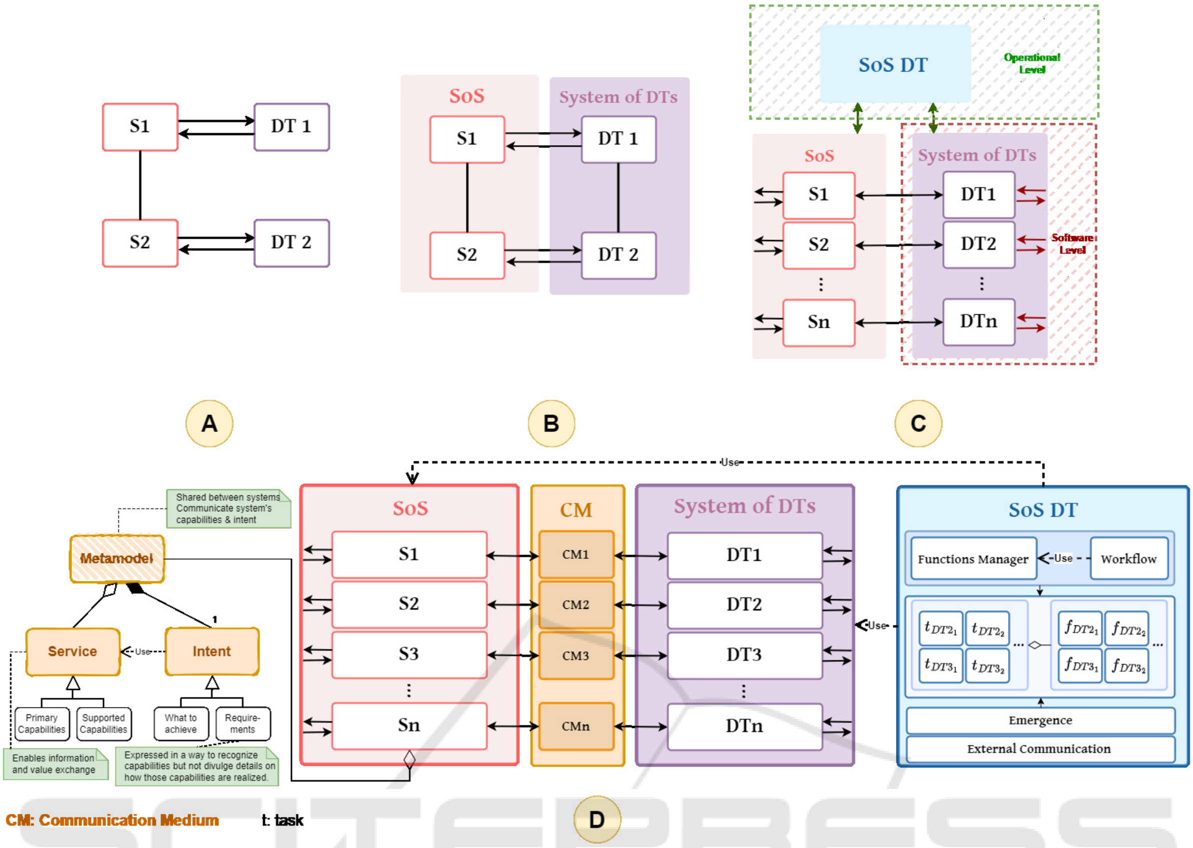

Figure 1: A. Communication Between Two Systems and Their Corresponding Digital Twins — B. A Simple SoS Composed of Two CSs Communicating with the *System of DTs* — C. SoS DT Architecture Proposal — Detailed SoS DT Architecture.

- Are only the permitted resources for sharing granted ?
- Is a specific type of communication medium adequate to represent all physical communications?
- What if different data types are involved?

At this level of complexity, **a *system of DTs*, of Figure 1.B, alone cannot effectively represent the SoS DT**.

3.3 The Key Contribution Needed for Accurate Representation of a SoS (RQ1.3)

To answer the questions mentioned in the Subsection. 3.2 and the sub-question RQ1.3, a new layer is introduced which is the SoS DT as shown in Figure. 1.C.

The SoS is composed of multiple CSs each potentially engaging in either unidirectional or bidirectional communication with other CSs. In this setup, each CS interacts directly with its corresponding DT. When the communication, on a software level, is established, each DT becomes a constituent DT of the *system of DTs*. However, due to the complexity of such systems and to fully address previously mentioned limitations, i.e resource constraints, communication mediums, and data diversity, an additional overarching layer, the SoS DT, becomes necessary.

The SoS DT serves as the "master," "manager," or "orchestrator" of the *system of DTs*, overseeing and coordinating interactions to ensure effective and cohesive functioning across the entire SoS. This orchestration layer provides an integrated control mechanism, facilitating streamlined data flow, efficient resource allocation, and synchronized operations within the SoS.

4 ENSURING ALIGNMENT OF THE PROPOSED ARCHITECTURE WITH CORE SoS CHARACTERISTICS (RQ1.4)

Aside from the indisputable characteristics mentioned by Maier in (Maier, 1998) (operational and managerial independence), other characteristics have been mentioned by (Andrew and Christopher, 2001) that the SoS DT needs to validate as well to be considered the replica of a physical SoS, these additional characteristics are: geographic distribution, evolutionary development and emergent behavior. To verify this matter, let us introduce the more detailed version of the proposition presented in Figure.1.D.

As explained previously, the SoS represents the communication among various existing or newly established systems. This communication on the physical side is governed by contractual agreements between the involved CSs. This agreement is depicted on the left side of the Figure.1.D wherein (Harbor and Research, 2024):

- A system conforms to a metamodel which is a framework that defines the standard structure and behavior for describing services, intents, and capabilities within a system. It ensures consistency and interoperability across different components and systems in a SoS.
- A service that is shared via the metamodel is a discrete unit of functionality offered by a system that can be consumed by other systems. It is modular and designed to perform specific tasks within the broader system.
- An intent refers to the desired outcomes or goals that a service aims to achieve. It guides the service in terms of what needs to be accomplished without specifying how it should be done. It ensures as well the alignment with the SoS objectives.
- Primary capabilities are the core tasks that a service provides to achieve its intent.
- Supported capabilities are the additional functionalities that a service can perform to enhance its primary capabilities.

We add to this contract that CSs need to adhere to the requirement of having a DT so that the creation of a *system of DTs* would be possible, thanks to the established communication medium that allows a bidirectional flow between the CSs and constituent DTs. Moving to the SoS DT that operates on the Operational Level that uses both the SoS and the System of DTs, that work on the Software level, to accomplish that master role it has. To do so, it is based on a workflow module that identify and extract tasks necessary to achieve the SoS's global objectives. And each task is composed of several different functions. This interoperability between tasks of constituent DTs give emergence of new tasks and are integrated to the SoS DT since they collaborate to the creation of the global goal of the SoS. The concept of external communication is illustrated to represent the interactions among multiple SoSs and their respective SoS DTs, but this aspect is beyond the scope of the present paper.

Following the explanation of Figure 1.D, a verification of the SoS criteria, presented in (Andrew and Christopher, 2001), is conducted to assess whether these criteria are also satisfied within the SoS DT framework:

- **Geographic Distribution:** Systems are often geographically dispersed, operating over wide areas and interacting through networks and communication systems. This criterion is verified in the SoS DT and the SoS due to the modular proposed architecture on both physical and digital sides (*system of DTs*).
- **Evolutionary Development:** Systems can evolve independently, allowing upgrades, modifications, or replacements without impacting the overall SoS. This adaptability is supported by operational and managerial independence in both twins verified previously.
- **Emergent Behavior:** The SoS exhibits behaviors and capabilities that emerge from the interactions between CSs, providing greater functionality than the systems could achieve individually. This concept of emergence is reflected in the two level of communications (software level, operational level) and is explicitly presented by the module ”Emergence” in the SoS DT.

5 THEORETICAL ILLUSTRATIONS

To illustrate and clarify the proposition, let us assume that the SoS DT is applied for resilience purposes. Here are some potential scenarios and theoretical use cases:

- *Scenario 1 – Failover Mechanism in Response to System Outage:* Within a SoS framework, unexpected failures of CSs can impact the overall performance and resilience. In such cases, the SoS DT autonomously detects disruptions through real-time data analysis and triggers a failover protocol by engaging the corresponding DTs to replace the CSs down allowing the performance to stay at a ”stable” level.
- *Scenario 2 – Response to Unknown Disturbance Impact:* Assuming a disturbance has occurred, but its impact on the overall performance of the SoS is unclear. In response, the SoS DT engages concerned constituent DTs, guiding them on appropriate actions, i.e whether to absorb, adapt, or re-

cover (Francis and Bekera, 2014), to uphold the performance.

- *Scenario 3 – Adaptive Response to Environmental Changes:* In complex systems such as a smart city SoS, CSs (e.g. traffic management, public transportation, emergency services) must adapt to environmental changes such as severe weather conditions. The SoS DT continuously monitors environmental parameters and communicates with the *System of DTs* to allow them help the physical SoS adapt its behavior in response.

6 CONCLUSION

SoS DTs represent a promising frontier in advancing the monitoring and control capabilities for complex, multi-system environments, specifically SoS. By establishing a virtual counterpart for each CS within an SoS, DTs enable comprehensive real-time insights, improving operational decision-making and performance management. However, the challenges associated with SoS DTs are significant. The SoS DT architecture showcases the need for modularity and scalability. Since each CS can evolve independently, added or replaced without disrupting the overall SoS, the digital side must accommodate to these changes seamlessly. A multi-DT approach, i.e each CS has its respective DT, provides a solution for managing this complexity, but maintaining the synchronization between the SoS DT, the *System of DTs* and the SoS, could be seen as a potential challenge since it is important to accurately replicate the SoS behavior.

Another challenge could be the interoperability between constituent DTs of the *system of DTs* as well as the communication between the *system of DTs*. Each CS may operate on different protocols, use various data formats, and have unique communication requirements. In a ”perfect” DT scenario, interoperability is assumed to function without interference or error. Nevertheless, practical implementations often face challenges related to data integration, compatibility between heterogeneous systems, and network latency. Addressing these issues requires robust communication standards and protocols that facilitate seamless data exchange while ensuring the fidelity of information being transmitted across the SoS.

In this paper, we assumed a ”perfect” DT scenario, where disturbances affect only the physical SoS. This idealized view highlights the potential of SoS DTs while underscoring the need for robust architectures capable of addressing real-world communication and interoperability issues.

Future work will focus on adapting the architecture for resilience matters and addressing the aforementioned challenges through rigorous testing on several real use cases. This is essential to uncover potential issues related to implementation, integration, and operation. By doing so, we can also evaluate whether the proposed architecture is generic enough to be applicable across various use cases, as DTs are often driven by specific use-case requirements (Göllner et al., 2022).

REFERENCES

Andrew, S. and Christopher, C. (2001). On the systems engineering and management of systems of systems and federations of systems. *Journal of Systems Engineering*, 21(4):325–345.

Azari, M. S., Flammini, F., and Santini, S. (2022). Improving resilience in cyber-physical systems based on transfer learning. In *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 203–208. IEEE.

Bhowmik, S. (2019). Digital twin of subsea pipelines: conceptual design integrating iot, machine learning and data analytics. In *Offshore Technology Conference*, page D011S010R004. OTC.

Borth, M., Verriet, J., and Muller, G. (2019). Digital twin strategies for sos 4 challenges and 4 architecture setups for digital twins of sos. In *2019 14th annual conference system of systems engineering (SoSE)*, pages 164–169. IEEE.

Broo, D. G., Bravo-Haro, M., and Schooling, J. (2022). Design and implementation of a smart infrastructure digital twin. *Automation in construction*, 136:104171.

Checkland, P. B. (1999). *Systems Thinking, Systems Practice*. John Wiley & Sons Ltd, Chichester, UK.

DeLaurentis, D. (2005). Understanding transportation as a system-of-systems design problem. In *43rd AIAA aerospace sciences meeting and exhibit*, page 123.

Demir, O. E., Colledani, M., Teicher, U., Achour, A. B., Seidel, A., Hänel, A., Yemane, A. T., Borzi, G. P., Uhlenkamp, J.-F., Eberlein, S., et al. (2023). Vertically integrated digital twins for rapid adaptation of manufacturing value chains. In *ECMS*, pages 435–441.

Francis, R. and Bekera, B. (2014). A metric and frameworks for resilience analysis of engineered and infrastructure systems. *Reliability engineering & system safety*, 121:90–103.

Grieves, M. and Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary perspectives on complex systems: New findings and approaches*, pages 85–113.

Göllner, D., Rasor, R., Anacker, H., and Dumitrescu, R. (2022). Collaborative modeling of interoperable digital twins in a sos context. *Procedia CIRP*, 107:1089–1094.

Harbor and Research (2024). System of systems. Accessed: 2024-11-07.

Josifovska, K., Yigitbas, E., and Engels, G. (2019). A digital twin-based multi-modal ui adaptation framework for assistance systems in industry 4.0. In Kurosu, M., editor, *Human-Computer Interaction. Design Practice in Contemporary Societies*, pages 398–409, Cham. Springer International Publishing.

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, 1(4):267–284.

Olsson, T. and Axelsson, J. (2023). Systems-of-systems and digital twins: A survey and analysis of the current knowledge. In *2023 18th Annual System of Systems Engineering Conference (SoSe)*, pages 1–6. IEEE.

Parri, J., Patara, F., Sampietro, S., and Vicario, E. (2021). A framework for model-driven engineering of resilient software-controlled systems. *Computing*, 103(4):589–612.

Pickering, N., Duke, M., and Au, C. K. (2023). Towards a horticulture system of systems: A case study of modular edge ai, robotics and an industry good digital twin. In *2023 18th Annual System of Systems Engineering Conference (SoSe)*, pages 1–8. IEEE.

Rauch, L. and Pietrzyk, M. (2019). Digital twins as a modern approach to design of industrial processes. *Journal of Machine Engineering*, 19.

Tan, Y., Yang, W., Yoshida, K., and Takakuwa, S. (2019). Application of iot-aided simulation to manufacturing systems in cyber-physical system. *Machines*, 7(1).