Digital Twins for Autonomous Intelligent Systems: From Development to Deployment

Lukas Esterle, Henrik Ejersbo, Mirgita Frasheri, Cláudio Gomes, Hugo Daniel Macedo, Peter Gorm Larsen
Department of Electrical and Computer Engineering and DIGIT
Aarhus University
Denmark
lukas.esterle@ece.au.dk

I. INTRODUCTION

Autonomous systems are operating without human intervention. This requires them to have a good model of themselves and their environment to ensure they operate safely towards their goals. Within a confined or static environment, a simple model can be enough to achieve this requirement. However, if we intend to have autonomous systems to operate intelligently within as well as together with their environment, a model alone will not suffice. Limitations and assumptions of the models and the initial prototypes that informed the modelling process, as well as environments never encountered while in prototyping stage, eventually render the model of a deployed system inaccurate. We can overcome this issue by updating the model at runtime. This is considered a digital shadow as it will reflect the current state of the environment at runtime. When this digital model is not only updated at runtime and reflects the current state but also is able to provide feedback to the real system at runtime, we consider this a Digital Twin (DT). Over the past years, DTs have become not only an essential tool in the development of cyber-physical systems but also an integral part of intelligent autonomous systems.

In this short tutorial paper and the corresponding ACSOS tutorial, we will discuss and work through the following topics:

- differences between computational models, digital shadows, and digital twins. We will further dive into the benefits and challenges of the different types of digital representations.
- development process facilitating digital twins to make better products.
- techniques to couple and combine different model simulations in order to enable digital twins
- the benefits of and challenges of having digital twins as part of an intelligent system. Even more so, discuss what is required to utilise digital twins in collaborative systems that have not been designed explicitly to operate with each other.

II. DIGITAL TWINS

Cyber-physical systems closely interact with the environment and various challenges arise for developers and engineers designing the different algorithms needed [1]. In the early 2000s, Grieves proposed Digital Twins (DTs) as digital replica of real systems to reduce the development costs and enhance testing and evaluation capabilities [2]. Importantly, normal computer models are inherently different as they reflect only a subset of the originals features and can be invalidated when the physical system is being deployed [3]. In contrast, DTs are not only being used during design time but continue to be used during deployment of the physical system. Here we can further distinguish between Digital Shadows and Digital Twins, where the former will only receive data from the physical twin and is often only used to monitor the physical counterpart. The latter, on the other hand, also feeds back information to the physical twin and allows for an interplay between the physical system and the digital replica. These three different levels have been illustrated in Fig. 1.

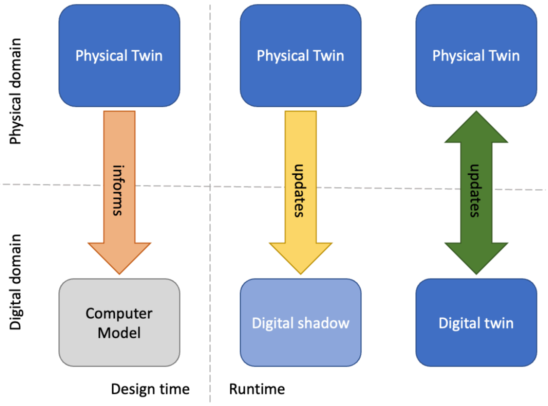

Fig. 1. Concept of Digital Twins.

III. CO-SIMULATION FOR DIGITAL TWINS

In order to create DTs, we have to create digital replicas of cyber-physical systems. To avoid building a full model for each system from scratch and facilitate re-use of models of individual components, co-simulation is utilised [4], [5]. Here, the individual models, often encapsulated with the open standard Functional Mock-up Interface as Functional Mock-up Units (FMUs), are simulated in unison in order to simulate the full system. In this tutorial, we will specifically look at the co-simulation toolchain called INTO-CPS [6]. The INTO-CPS toolchain allows to co-simulate models using different modelling approaches. This enables developers to analyse and explore different implementations of their system in simulation before building the physical counterpart. During the development cycle, individual components of the co-simulated model can be replaced with hardware components, offering all benefits of hardware-in-the-loop simulations within the INTO-CPS toolchain. At this stage, the simulated components will already interact with the physical counterparts, implementing a first stage of a digital twin.

IV. DIGITAL TWINS IN INTELLIGENT SYSTEMS

Within the last years, DTs have evolved from development and monitoring tools towards full system support and control tools. These DTs are able to interact and respond to changing conditions in the environment and give feedback to the CPS at runtime. In order to establish the communication between the digital and the physical twin, we developed a framework that allows us to couple the twins with a standardized message protocol technology. Specifically, we developed an FMU on top of the RabbitMQ open-source message-broker software, that enables bi-directional communication. Specifically, we developed an FMU for rabbitMQ to enable bi-directional communication between the digital twin and the physical counterpart [7], [8]. This allows sending information not only from the physical system to the DT but also send control signals the other way. For example, when a discrepancy between the DT and the physical twin is detected and safety guarantees are about to be violated, an emergency stop or evasive manoeuvre could be triggered in an autonomous vehicle in order to avoid harm to others or itself. Furthermore, this communication module facilitates information exchange among other DT or components if necessary. Finally, we are able to encapsulate communication and test different faults either on the sensor and/or communication level [9].

DTs act as enablers for intelligent systems allowing the development of an awareness of their own competences. Only through such an awareness, any intelligent system can establish their full potential and strive to improve beyond for what it was originally designed [10]. This can be achieved by integrating the DT within autonomous control loops (e.g., MAPE-K [11]). When multiple systems operate within the same environment, they have to coordinate themselves in order to ensure to not interfere with their respective objectives [12]. This leads to extensive communication overhead when all systems need to align their actions before the actual execution. An alternative can be envisioned when all these systems also utilise DTs in their operation. In that case, they can facilitate self-integration of different systems through their DTs [13]. Here, individual systems only exchange information about themselves using the DTs. Expected actions can be simulated by others by executing the respective DT. This can lead to an understanding of their mutual capabilities, goals, and potentially their respectively planned actions [**?**]. Using mutual execution of their DT models, both systems can get an understanding of the behaviour of the other system. Either one of them can utilise this knowledge to plan and prepare its own next steps. Using consensus finding approaches, the individually planned collaborations can be harmonised and individual as well as common goals can potentially be achieved more effectively. With enough advancement towards self-integration, we will end up with new DTs of systems-of-systems. These again can be utilised to verify the overall state of all systems [14]. Through this we aim to overcome epistemic uncertainties, i.e. having insufficient information, and aleatoric uncertainties, i.e. having inherent randomness in the available information, of the verification process.

ACKNOWLEDGMENT

The work in this tutorial was funded in parts by the Poul Due Jensen Foundation, the Innovation Foundation Denmark and ITEA for funding the AgroRobottiFleet project and the UPSIM project.

REFERENCES

[1] L. Esterle and R. Grosu, "Cyber-physical systems: challenge of the 21st century," *e & i Elektrotechnik und Informationstechnik*, vol. 133, no. 7, pp. 299–303, 2016.

[2] M. W. Grieves, "Product lifecycle management: the new paradigm for enterprises," *International Journal of Product Development*, vol. 2, no. 1-2, pp. 71–84, 2005.

[3] T. Kühne, "What is a model?" 2005.

[4] J. Fitzgerald, K. Pierce, and P. G. Larsen, "Co-modelling and co-simulation in the engineering of systems of cyber-physical systems," in *2014 9th International conference on system of systems engineering (SOSE)*. IEEE, 2014, pp. 67–72.

[5] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–33, 2018.

[6] P. G. Larsen, J. Fitzgerald, J. Woodcock, C. König, S. Basagiannis, E. Brosse, C. Gomes, J. Cabral, H. D. Macedo, C. Thule, A. Sadovykh, C.-B. Zamfirescu, M. Neghina, K. Pierce, C. Gamble, and R. Payne, "The INtegrated TOolchain for Cyber-Physical Systems (INTO-CPS): a Guide," INTO-CPS Association, Tech. Rep., 2018.

[7] C. Thule, C. Gomes, and K. G. Lausdahl, "Formally verified fmi enabled external data broker: Rabbitmq fmu," in *Proceedings of the 2020 Summer Simulation Conference*, ser. SummerSim '20. San Diego, CA, USA: Society for Computer Simulation International, 2020.

[8] M. Frasheri, H. Ejersbo, C. Thule, and L. Esterle, "Rmqfmu: Bridging the real world with co-simulation technical report," *arXiv preprint arXiv:2107.01010*, 2021.

[9] M. Frasheri, C. Thule, H. D. Macedo, K. Lausdahl, P. G. Larsen, and L. Esterle, "Fault injecting co-simulations for safety," in *2021 5th International Conference on System Reliability and Safety (ICSRS)*. IEEE, 2021, pp. 6–13.

[10] L. Esterle and J. N. Brown, "The competence awareness window: Knowing what i can and cannot do," in *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. IEEE, 2020, pp. 62–63.

[11] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[12] K. Bellman, J. Botev, A. Diaconescu, L. Esterle, C. Gruhl, C. Landauer, P. R. Lewis, P. R. Nelson, E. Pournaras, A. Stein *et al.*, "Self-improving system integration: Mastering continuous change," *Future Generation Computer Systems*, vol. 117, pp. 29–46, 2021.

[13] L. Esterle, C. Gomes, M. Frasheri, H. Ejersbo, S. Tomforde, and P. G. Larsen, "Digital twins for collaboration and self-integration," in *2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. IEEE, pp. 172–177.

[14] L. Esterle, B. Porter, and J. Woodcock, "Verification and uncertainties in self-integrating system," in *2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. IEEE, pp. 220–225.