



HAL
open science

A software engineering point of view on digital twin architecture

Gaëlic Bechu, Antoine Beugnard, Caroline Gl Cao, Quentin Perez, Christelle Urtado, Sylvain Vauttier

► **To cite this version:**

Gaëlic Bechu, Antoine Beugnard, Caroline Gl Cao, Quentin Perez, Christelle Urtado, et al.. A software engineering point of view on digital twin architecture. ETFA 2022 - IEEE 27th International Conference on Emerging Technologies and Factory Automation, Sep 2022, Stuttgart, Germany. 10.1109/ETFA52439.2022.9921617 . hal-03832378

HAL Id: hal-03832378

<https://imt-mines-ales.hal.science/hal-03832378>

Submitted on 28 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A software engineering point of view on digital twin architecture

Gaëlic Béchu¹, Antoine Beugnard¹, Caroline G. L. Cao¹,
Quentin Perez², Christelle Urtado², Sylvain Vauttier²

¹LabSTICC, IMT Atlantique, Brest, France

¹{firstname.lastname}@imt-atlantique.fr

²EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France

²{firstname.lastname}@mines-ales.fr

Abstract—Digital twins, along with Internet of Things and Artificial Intelligence, have been identified as one of the key technologies for Industry 4.0. However, the definition of Digital Twin (DT) is still abstract and context-dependent. In this paper, we present a metamodel that supports concrete and operational descriptions of digital twin deployment. This metamodel encompasses the different aspects of deployment, including the definition of hardware and software components that compose the layered cyber-physical architectures of the digital twin, along with the installation and instantiation tasks that compose deployment processes. Multiple configurations can also be defined to support the deployment of a digital twin in different execution contexts. The relevance of this metamodel was evaluated by two case studies. The first consists of deploying the digital twin of a cobot in a simulation environment. The second applies the approach in a home automation environment. In both cases, our metamodel provides complete and precise descriptions of the deployment process and thus constitutes a viable first step towards a model-driven approach for digital twin deployment.

Index Terms—Digital Twin, Model-Driven Engineering, Architecture Deployment.

I. INTRODUCTION

Digital twins, along with Internet of Things and Artificial Intelligence, have been identified as one of the key technologies for Industry 4.0. However, as for many emerging concepts, the definition of Digital Twin is still abstract and context-dependent. We present in this paper a metamodel that supports concrete and operational descriptions for digital twin deployment. The remainder of this paper is organized as follows. Section II details the metamodel we proposed to handle the deployment of digital twins in a model-driven approach. Section III presents an evaluation of the completeness and adaption of our metamodel in two case studies. Section V discusses these preliminary results and provides perspectives about this on-going work.

II. A METAMODEL FOR MODEL-DRIVEN DEPLOYMENT OF DIGITAL TWINS

If there is no consensus on the definition of Digital Twins, there is even less agreement on their software architecture. The ISO23247 standard [1] provides a global architecture where “boxes” communicate through connecting “lines”. These boxes are:

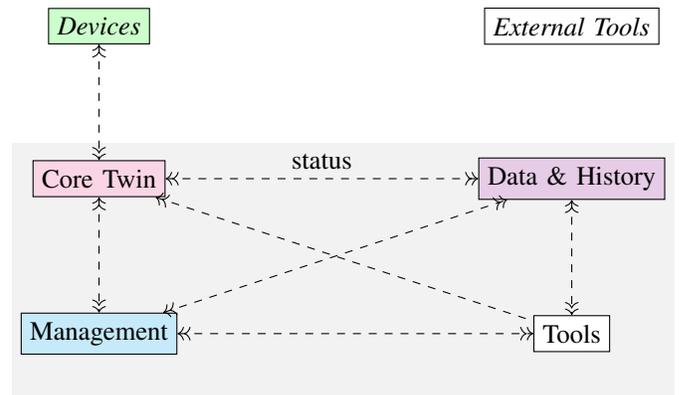


Fig. 1: Digital Twin macro-architecture

- User Entity including Human, Human-Machine Interaction, Manufacturing Execution Systems, Enterprise Resource Planning and other DTs,
- Core Entity including Management Services and Digital Representation of Observable Manufacturing Elements,
- Data Collection and Device Control Entities that gather elements to connect actual devices,
- Cross System Entity including all communication services.

This standard is not a software architecture; it is a reference architecture. It includes real “things” such as devices, humans, and applications. The software part of DTs (yet to be defined) is complex. Among all software, DTs may use OSs, drivers, controllers of hardware, databases, data conversion, data storage, data analysis applications, AI tools, 3D engines, etc. Hence, defining a software architecture is context-dependent. This paper defines a synthetic architecture as a simple organization of software and data streams.

Figure 1 illustrates the adopted organization of our DTs. The overall architecture is composed of four main packages. The content is digital (i.e., files such as electronic documents, spreadsheets, software, scripts, text files, etc.)

The *Devices* box denotes the sociotechnical system (STS) itself, including humans, which are not part of the DT.

The *External Tools* box refers to the external software that can

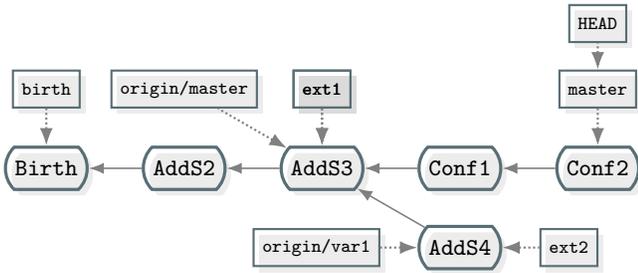


Fig. 2: A DT configuration: its birth, add some parts, two configurations and two variants (branches)

be used (and shared) and that usually requires configurations that are part of either the Core or the Tools boxes.

The *Core Twin* box contains everything (hardware description and software) defining the STS. It includes sensors, actuators and drivers. It also contains all brokers (scripts, software) ensuring the connection between all parts (software and data) including status extraction and data historization.¹

The *Data & History* box contains the status of the DT and all historical (biographical) data. It includes 3D-models, sensors values, actuators commands, etc. If, for some reason (software installation/configuration constraint), data have to be located elsewhere, a link is explicitly made².

The *Tools* box contains software such as simulation, analyzing or learning tools. It reads information from Data & History and feeds it with new information (statistics, configurations). Dependency among tools and data have to be explicit, as in a configuration management tool such as git (see Figure 2).

The *Management* box contains software, scripts and data used to manage the DT. It includes introspection services (request), GUIs, controllers, etc.

Beyond this organization, we define an ontology that gathers all needed concepts to describe, implement and manage a DT. An excerpt is shown in Figure 3. For the sake of readability, we used colors to refer to the architecture of Figure 1. The green part (Devices) is a hierarchy of *RealWorldThing* including *Hardware* and *LivingBeing*. The red part represents the core digital twin. The blue part describes management with *Services* and *ConfigurationScript*. The purple part denotes Data produced and consumed by *Software*. The white part represents *Tools*, whereas the yellow one denotes abstractions that organize concepts.

III. MODELING THE DEPLOYMENT OF DIGITAL TWINS IN TWO INTELLIGENT ENVIRONMENTS

A. Case Study #1: Human-Robot Interaction

The first deployment case study focuses on the collaboration between a robotic arm and its operator while they are performing a collaborative task. This case study mimics the environment of a worker in a manufacturing plant and was created

¹ Configuration data splits into 2 categories: technical and model. Model data must be located in the Data & History component, while technical must be in the Core Twin. Technical data collect logins, IP addresses, etc.

² For example, a Unix symbolic link, or some kind of readme file at least.

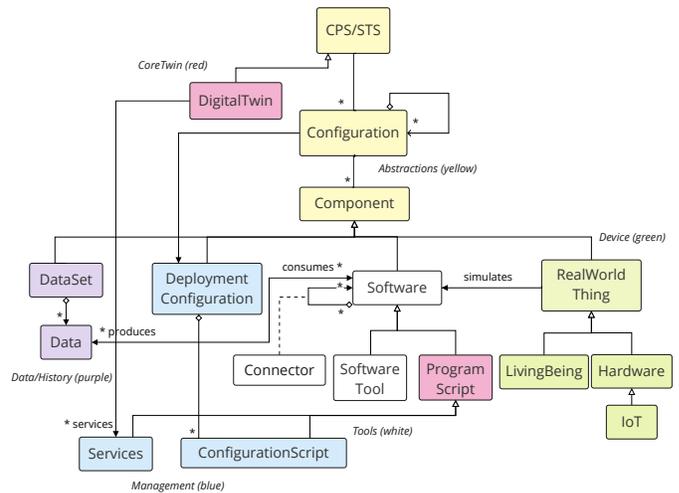


Fig. 3: A DT ontology (excerpt)

with the purpose of joining through a DT a human operator and a robotic arm. The digitalization of this interaction is done with Unity³ which is the simulator that best suits our needs. The robotic arm at our disposal is controlled on ROS⁴. We could have remained with a ROS-friendly simulator such as Gazebo, but while being adequate for robotics applications, Gazebo is not ideal for the digitalization of anything non-robotic [2]. We created the DT of the Human using gyroscopes placed on different body parts of our operator. We streamed this data between different computers to have a DT of our human operator in Unity. The factor to consider here is the communication between the various hardware and software tools. These tools were designed to work on their own for different purposes and we had to build a generalized architecture that integrated different communication means (Radio, TCP/IP, USB). We built a Composite Structure Diagram (CSD) according to the ontology proposed in Section II. As shown in Figure 4, the different hardware components support different applications and software tools. We can see the communication between the hardware components as well as the structure of the DT. The Human-Robot Interaction DT can be seen as the combination of two different DTs: the Human DT and the Robot DT.

B. Case Study #2: Home Automation System

The second case study concerns a Home Automation System (HAS). The HAS is composed of two major elements: hardware (sensors, actuators, computers, routers, etc.) and software (home automation software, operating systems, etc.) In this case, we mix real (smart light bulb, computers, router, etc.) and simulated hardware. We built a predictor of a bed pressure sensor which indicates whether a person is in bed or not. To do so, we trained a forecasting model (naive forecaster) to learn data patterns with Sktime. Data used for learning come from the dataset of Chimamiwa *et al.* [3]. We then integrated this simulated sensor in the HAS.

³ <https://unity.com/>

⁴ <http://wiki.ros.org>

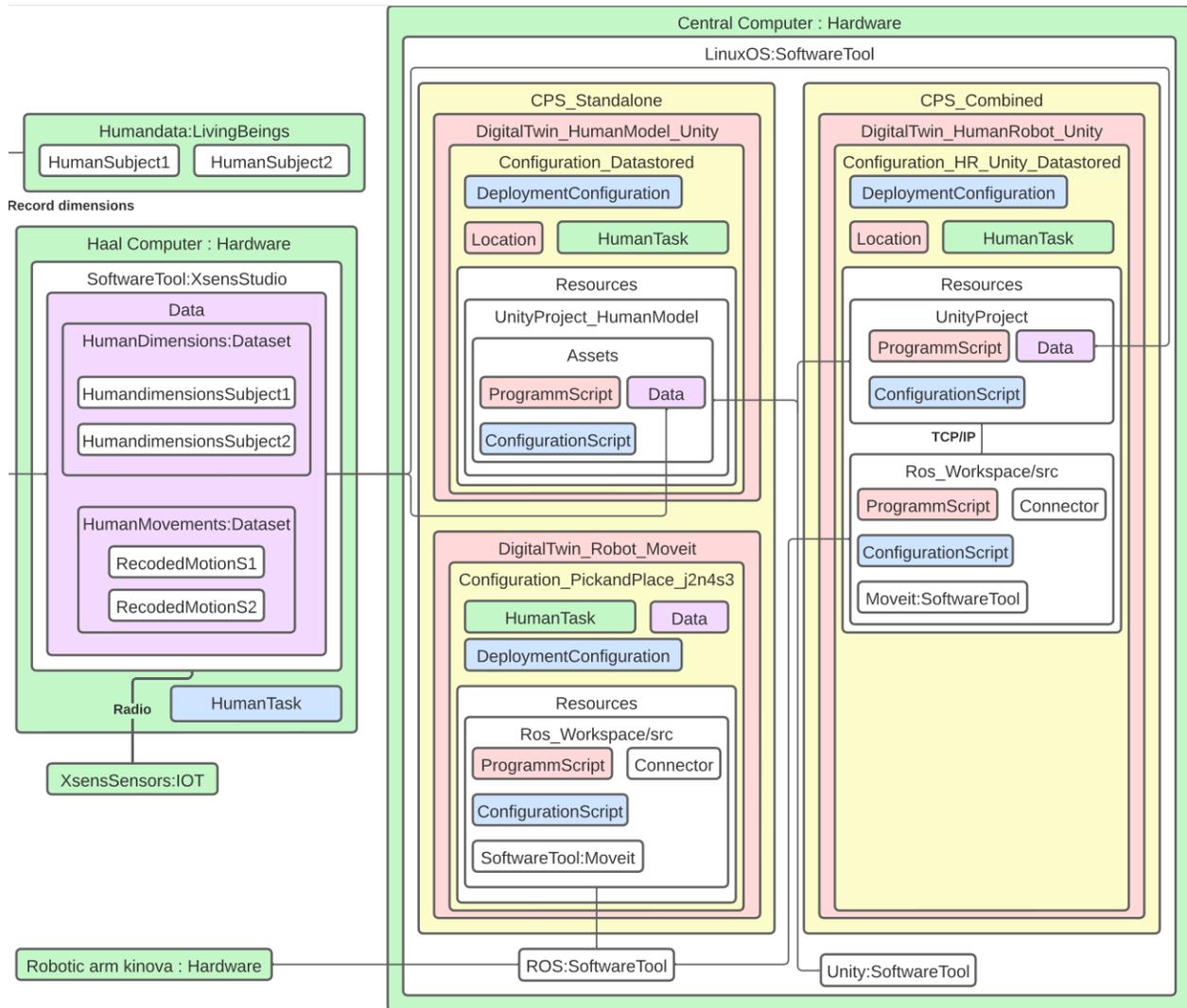


Fig. 4: CSD of a Human Robot Interaction collaborative task using the DT ontology

Next, we built a Composite Structure Diagram of the HAS using our proposed ontology described in Section II. As shown in Figure 5 we are able to take into consideration all the hardware and software composing the HAS. Moreover, the ontology allows the composition of Cyber-Physical System (CPS). In Figure 5, it is the case of HomeAutomationSystem and BedPressureSystem. HomeAutomationSystem is the CPS composed of a Raspberry Pi 4 and the dedicated OS installed on it. Other components such as the zWaveDongle are also plugged in. On the Raspberry Pi OS, the home automation application (*i.e.*, OpenHAB⁵) is installed. The ontology also allows the modeling of components and technologies to communicate between devices and the CPS. The Z-Wave protocol is used for the interaction between the light bulb and the OpenHAB software. Another example in Figure 5 is the WiFi communication between the two CPSs through the router.

⁵ <https://www.openhab.org/>

A prospective work is to use this approach to model and manage DTs for the connected apartment of the HUMAN at home project (HUT)⁶ and use them as a workbench for the development and training of AI-based mechanisms.

IV. RELATED WORK

For the sake of space, we reference a survey from Minerva *et al.* [4] which draws an interesting picture of technical features of DTs and shows that most software architectures published are tailored to specific goals. Taking a different approach, our proposal provides a generic, model-driven solution for constructing and deploying DT architectures.

V. DISCUSSION AND CONCLUSION

We have performed a preliminary validation of our metamodel defined in Section II with two different use cases. The first one is a human-robot interaction (HRI) system. The second is

⁶ <https://www.hut-occitanie.eu/>

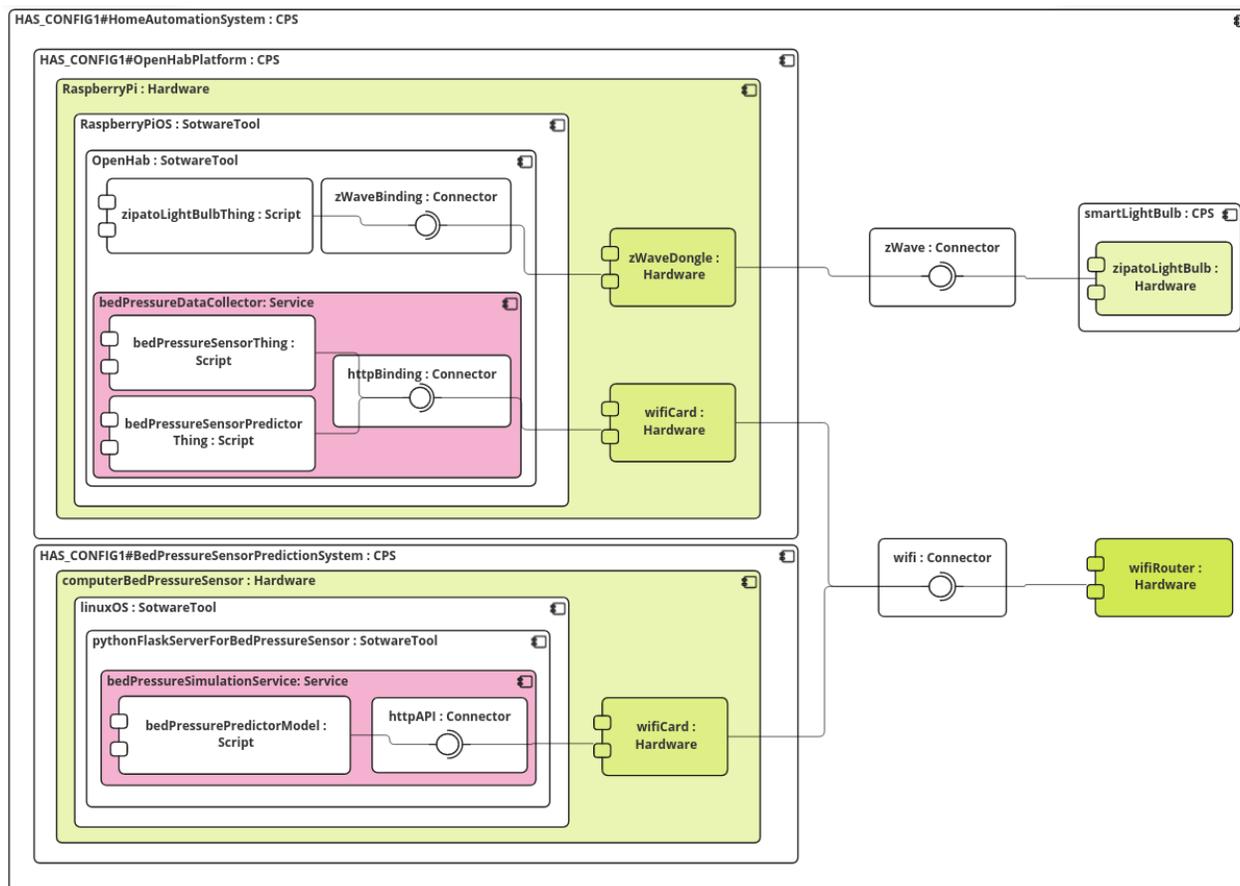


Fig. 5: CSD of the Home Automation System using the DT ontology.

a smart environment system. In this work-in-progress report, we propose our ontology as a generic tool for modeling DT architectures.

We are continually improving our models and verifying them with practical experiments. We aim to refine the definition of “digital twin”, initially by defining different categories according to different configurations. This categorization will allow us to identify the different shortcomings of these configurations.

Our focus on HRI digital twin brings multiple perspectives. We strive to understand the methodology needed to create a digital twin, while at the same time to create a dataset containing our HRI interactions. To date, many HRI datasets exist but are mainly focused on behavior analysis and not from a manufacturing point of view⁷. Our approach allows us to contribute to a comprehensive database for constructing digital twins for the future smart factories of Industry 5.0 [5]. Our second use case of a smart environment system showcases the general applicability of our approach. Daily activity datasets are also scarce and difficult to collect. Building digital twins could be a way to leverage the data collected in the HUT project after the dismantling of its smart apartment.

⁷ <https://github.com/mjyc/awesome-hri-datasets>

ACKNOWLEDGEMENTS

This work is funded in part by the CARNOT MINES and TSN Institutes, the Chair of Industry of the Future funded by the Région de Bretagne and University of South Australia, as well as GT VOCA of the HUMAN at home project (HUT) funded by FEDER and the Région Occitanie. We thank Panagiotis Papadakis, Christophe Lohr and Alireza Asvadi for their expertise.

REFERENCES

- [1] *ISO 23247-2:2021 Automation systems and integration - Digital twin framework for manufacturing - Part 2: Reference architecture*, International Standard Organization, 2021.
- [2] M. S. P. de Melo, J. G. da Silva Neto, P. J. L. da Silva, J. M. X. N. Teixeira, and V. Teichrieb, “Analysis and comparison of robotics 3d simulators,” *2019 21st Symposium on Virtual and Augmented Reality (SVR)*, pp. 242–251, 2019.
- [3] G. Chimamiwa, M. Alirezaie, F. Pecora, and A. Loutfi, “Multi-sensor dataset of human activities in a smart home environment,” *Data in Brief*, vol. 34, p. 106632, 2021.
- [4] R. Minerva, G. M. Lee, and N. Crespi, “Digital twin in the iot context: A survey on technical features, scenarios, and architectural models,” *PROCEEDINGS OF THE IEEE*, vol. 108, no. 10, pp. 1785–1824, October 2020.
- [5] D. Romero and J. Stahre, “Towards the resilient operator 5.0: The future of work in smart resilient manufacturing systems,” *Procedia CIRP*, vol. 104, pp. 1089–1094, 2021.