Enabling 3D Simulation in ThingsBoard: a First Step Towards a Digital Twin Platform

Massimo Callisto De Donato[1], Flavio Corradini[1] Fabrizio Fornari[1], Barbara Re[1], and Matteo Romagnoli[1]

University of Camerino, School of Science and Technology, Computer Science Department, Via Madonna delle Carceri, 7, Camerino, Italy
name.surname@unicam.it

Abstract. Digital twin platforms enable the creation, management, and analysis of digital twins. However, most of the available platforms are distributed as proprietary software. Considering that available digital twin platforms often generate from former IoT platforms and that visualization and simulation are among the main characteristics of digital twins, we present an extension we developed to the ThingsBoard open-source IoT platform to support the design and development of 3D simulations. Here, we describe such a simulation mechanism and its application on a smart classroom use case. Our work represents a first step towards an open-source digital twin platform.

Keywords: digital twin, digital twin platform, 3D simulation

1 Introduction and Motivation

A digital twin is a virtual representation of a physical object, process, or system. It uses real-time data and simulations to mimic the behavior, characteristics, and interactions of its real-world counterpart [5, 6]. The concept of a digital twin allows organizations to gain insights into the performance, maintenance, and optimization of physical assets and processes. It brings together various technologies, such as the Internet of Things (IoT), data analytics, artificial intelligence (AI), and cloud computing, to create a comprehensive and dynamic representation of the physical entity in the digital realm.

A digital twin platform is a technology solution that enables the creation, management, and analysis of digital twin models [8]. Essentially, Digital Twin platforms allow to combine data retrieved from the physical entity with the models that represent the physical entity. Several models can be defined to describe the physical entity such as *3D models*, *behavioral models*, *physical models*, *mathematical models*, etc. Through interactive controls, users can manipulate the digital counterpart, explore different perspectives, and gain a better understanding of its structure and functioning. In addition, those kind of models enable the possibility to conduct simulations to test various scenarios and identify potential issues before they occur in the physical world. In fact, by simulating the behavior of the physical entities, in different conditions and scenarios, users can predict how the entities will behave before implementing any change or making any decision. We report in Fig. 1 a conceptual representation of a Digital Twin Platform.

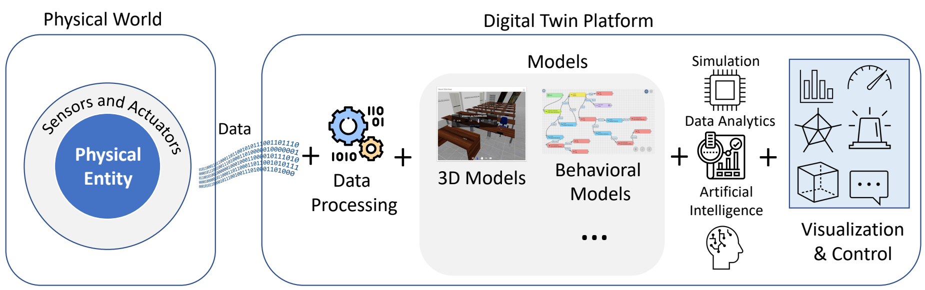


Fig. 1. Concept of a Digital Twin Platform

Several Digital Twin platforms have started to appear in the market such as *Azure Digital Twins*, *AWS IoT TwinMaker*, *iTwin Bentley*, and many others. They often generate from former IoT platforms and have different characteristics and provide different supports for Digital Twins [8, 7]. However, to the best of our knowledge, a comprehensive open-source DT platform is not made available yet [4]. It is towards the development of such a DT platform that in this work we present an extension of the ThingsBoard IoT platform[1], to start integrating some of the main features of Digital Twins, especially referring to the support for 3D simulation. As recognized by recent works, ThingsBoard is among the most well-known open-source platforms for IoT applications [3, 2], it provides a powerful set of tools for building and managing connected devices and applications. The platform implements a wide range of functionalities (device integration, big data storage, data processing, visualizations) defined to support both developers and businesses to quickly create scalable IoT solutions without the need for extensive coding. Thingsboard has been recognized as a good candidate for supporting digital twins [1]. Especially, it has been compared with other platforms and it emerged that ThingsBoard already implements a good portion of digital twin characteristics, but with a limited support for visualization and simulation [1]. Therefore, we extended ThingsBoard by adding the possibility to support and visualize 3D models associated with digital entities registered in the platform and with the possibility to design and execute 3D simulations.

The rest of the paper is structured as follows. Section 2 describes in detail the 3D simulation mechanism we defined. Section 3 describes the steps necessary to design a 3D simulation. Finally, Section 4 closes by reporting concluding remarks.

[1] https://thingsboard.io/

2 3D Simulation with ThingsBoard

The 3D simulation design and execution rely on a *simulation widget* we developed. The widget allows to create and simulate real-world scenarios and analyze the behavior of IoT systems in a virtual environment. ThingsBoard widgets[2] are additional UI modules that easily integrate into any dashboard that can be created in ThingsBoard. They provide end-user functions such as data visualization, remote device control, alarms management, and display of static custom HTML content. ThingsBoard allows also the development and integration of customized widgets that add functionalities to the platform. Fig. 2 represents the 3D simulation widget with the various components involved in the design and execution of a 3D simulation.

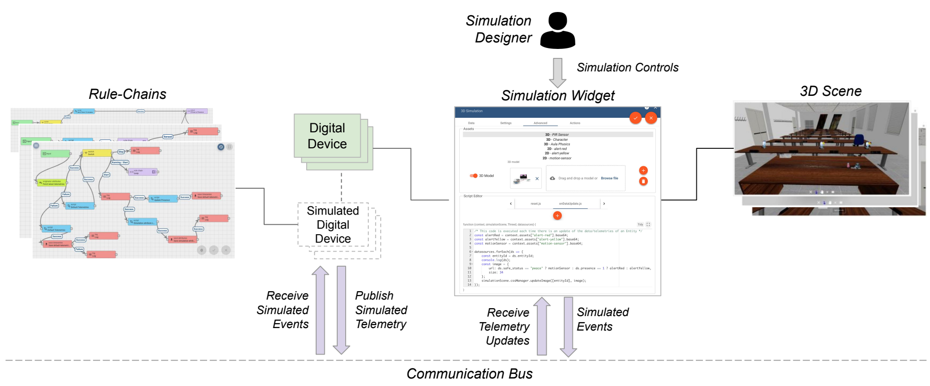


Fig. 2. Schematization of the 3D Simulation Mechanism.

ThingsBoard has a mechanism to define digital representations of IoT devices, we refer to them as *Digital Devices*. Such digital devices are enriched with attributes, treated as key-value pairs, that describe characteristics of the physical device such as *name*, *description*, *firmware version*, *latitude*, *longitude*, etc. In addition, telemetry data coming from the physical devices can be associated with the digital devices, i.e., in the case of an HVAC system such telemetry data can regard *temperature*, *humidity*, *status* (on/off), etc.

Among the components required to design and run a 3D simulation, we also included *Simulated Digital Devices*, see Fig. 2. Those devices inherit all the characteristics of digital devices, but they are the ones actually used for running the simulations. This distinction is required to avoid simulated telemetry data from overwriting real telemetries coming from the physical world and reflected on the digital device.

For creating a 3D simulation, the *Simulation Designer* needs to specify the digital devices and the simulated digital devices that he would like to involve

[2] https://thingsboard.io/docs/user-guide/ui/widget-library/

in the simulation. Then, by means of the 3D simulation widget, the user can design a 3D simulation including the digital devices reported on the ThingsBoard platform. Some familiarity with the Three.js[3] library used for designing the *3D Scene* as well as the cannon-es[4] library used for handling the physics is required.

The device behavior is simulated by means of the ThingsBoard *rule engine* which is a tool for processing and analyzing data generated by physical devices and associated with digital ones. It allows users to define complex rules, named *Rule-Chains* in terms of connected control flows where certain conditions can trigger specific actions based on the data received. In our case, we proposed adopting such a tool for encoding the behavior of a digital device and associating such behavior to a simulated digital device.

When a simulation is activated by a user, simulated events in the simulated environment may occur. Such simulated events are published on a *communication bus* and received by the corresponding simulated digital devices which handle the event by updating their telemetry. If a rule-chain that predicates on that event is available then the rule-chain fires and the simulated behavior of the simulated digital device starts. Also, the execution of a rule-chain might cause the update of some telemetries associated with the simulated digital device. Such telemetries are then published on the communication bus and received by the simulation widget that will reflect those updates in the simulated environment.

3 Use case

We used ThingsBoard and the 3D simulation mechanism previously introduced to simulate a smart classroom scenario inspired by the SAFE project [9] which involves smart furniture equipped with PIR devices for the detection of people in case of a seismic event.

By following the mechanism defined in Sec. 2 we first created the digital devices that represent the physical PIR devices of the SAFE scenario (*PIR-1*, *PIR-2*, and *PIR-3* in Fig. 3 part-a). Then, we defined the corresponding simulated digital devices (*SIM-PIR-1*, *SIM-PIR-2*, and *SIM-PIR-3*) to which we associated the simulated behavior encoded as ThingsBoard rule-chains.

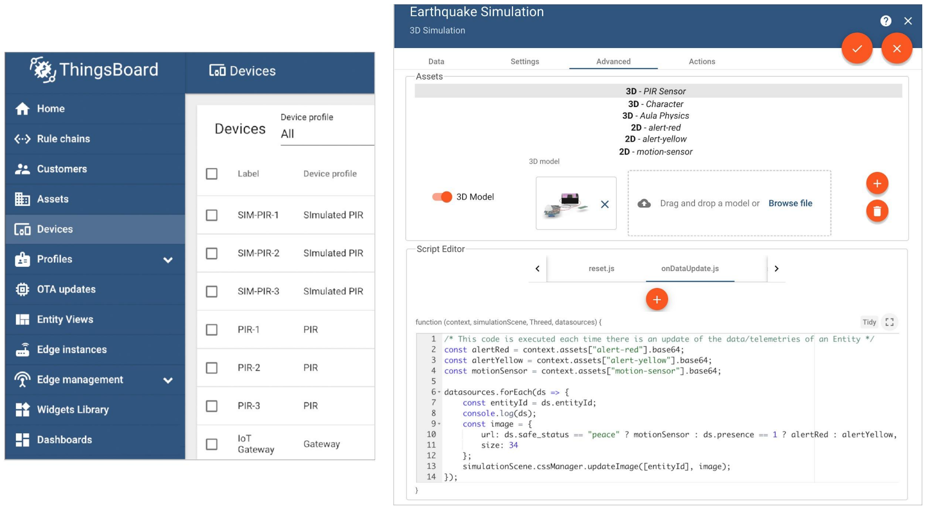
To instantiate the widget, we followed the ThingsBoard's standard approach which required us to create a new dashboard specifying the digital devices we wanted to use in the simulation. Then, we added the new instance of the 3D simulation widget and we started the design process. In Fig. 3 part-b, we reported the widget configuration that we defined acting as *simulation designers* which includes the import of the 3D models[5] and the coding of the simulation scenario.

Using the 3D models of the devices and of the environment these devices populate (a classroom), we designed, by means of scripting, a 3D scene like the one reported in Fig. 4 and we programmed parts of its behavior to drive the

[3] https://threejs.org/

[4] https://pmndrs.github.io/cannon-es/

[5] 3D models have been created by means of third-party tools such as Blender and exported in glTF format.



(a) (b)

Fig. 3. Design a 3D Simulation with ThingsBoard: a) reports a list of digital devices and simulated digital devices; b) reports the configuration of the 3D simulation widget.

simulation's execution. In our use case, as shown in Fig. 5, the scene we designed intends to simulate the occurrence of a seismic event showing the change in the behavior of the simulated PIR devices when the presence of a person taking shelter under the smart furniture is detected. This kind of simulation allows for rapid prototyping of possible classroom scenarios, envisioning different furniture dispositions or different positions of the PIR devices. Especially, the simulation enabled us to evaluate the PIR behavior by inspecting the rule chains execution, assessing whether the devices behaved as expected.

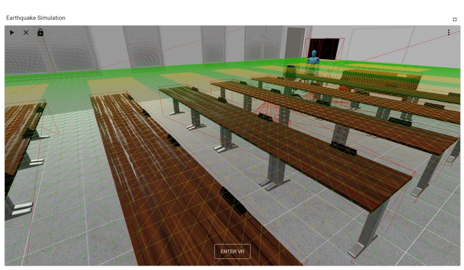


Fig. 4. 3D model of the scene

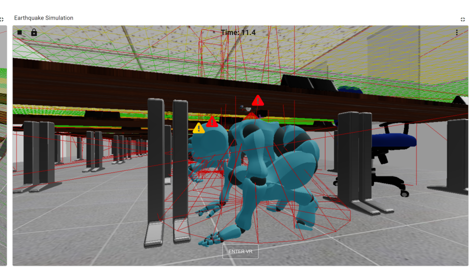


Fig. 5. Running 3D simulation

4 Conclusion and Future Work

We presented a 3D simulation mechanism for extending the ThingsBoard platform so to start its transition towards a digital twin platform. A user can design his own 3D simulation of a cyber-physical system, include the digital devices of interest, and start the simulation from real telemetry data. We have shown the steps necessary to implement a 3D simulation by means of a smart classroom use case. Currently, designing a graphical simulation requires programming skills, therefore we envision to define alternative approaches to facilitate this step. We also plan to conduct an evaluation with users. Further details and the source code are available at https://pros.unicam.it/digitaltwin/dtplatform. A screencast showing the design of the 3D simulation introduced in this paper is available at https://youtu.be/up2fwEMH7Vg.

Acknowledgements This work has been partially supported by the European Union – NextGenerationEU - National Recovery and Resilience Plan, Mission 4 Education and Research - Component 2 From research to business - Investment 1.5, ECS_00000041-VITALITY - Innovation, digitalisation and sustainability for the diffused economy in Central Italy.

References

1. Corradini, F., Fedeli, A., Fornari, F., Polini, A., Re, B.: DTMN a modelling notation for digital twins. In: Enterprise Design, Operations, and Computing. EDOC 2022 Workshops, Bozen-Bolzano, Italy, October 4-7, 2022. Lecture Notes in Business Information Processing, vol. 466, pp. 63–78. Springer (2022)
2. Corradini, F., Fedeli, A., Fornari, F., Polini, A., Re, B.: FloWare: a model-driven approach fostering reuse and customisation in IoT applications modelling and development. Software and Systems Modeling pp. 1–28 (2022)
3. Corradini, F., Fedeli, A., Fornari, F., Polini, A., Re, B., Ruschioni, L.: X-IoT: a model-driven approach to support IoT application portability across IoT platforms. Computing pp. 1–25 (2023)
4. Fei, T., Xuemin, S., CHENG, J., Yonghuai, Z., Weiran, L., Yong, W., Hui, X., Tianliang, H., Xiaojun, L., Tingyu, L., et al.: maketwin: A reference architecture for digital twin software platform. Chinese Journal of Aeronautics (2023)
5. Grieves, M.: Intelligent digital twins and the development and management of complex systems. Digital Twin **2**(8), 1–8 (2022)
6. Grieves, M., Vickers, J.: Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. Transdisciplinary perspectives on complex systems: New findings and approaches pp. 85–113 (2017)
7. Lehner, D., Pfeiffer, J., Tinsel, E., Strljic, M.M., Sint, S., Vierhauser, M., Wortmann, A., Wimmer, M.: Digital twin platforms: Requirements, capabilities, and future prospects. IEEE Softw. **39**(2), 53–61 (2022)
8. Pfeiffer, J., Lehner, D., Wortmann, A., Wimmer, M.: Modeling capabilities of digital twin platforms - old wine in new bottles? J. Object Technol. **21**(3), 3:1–14 (2022)
9. Pietroni, L., Mascitti, J., Galloppo, D., et al.: The SAFE project: an interdisciplinary and intersectoral approach to innovation in Furniture Design. DS 118: Proceedings of NordDesign 2022, Copenhagen, Denmark, 16th-18th August 2022 pp. 1–12 (2022)