Design and Development of a Digital Twin Prototype for the SAFE Project

Massimo Callisto De Donato[1][0000−0001−6745−4785], Flavio Corradini[1][0000−0001−6767−2184] Fabrizio Fornari[1][0000−0002−3620−1723], Barbara Re[1][0000−0001−5374−2364], and Matteo Romagnoli[1][0009−0001−5829−2996]

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

Abstract. The rapid advancements in digital technologies have paved the way for the development and utilization of digital twins that allow bridging the gap between physical systems and their virtual representations. This digital twin concept is gaining importance especially in the design of complex IoT and Cyber-Physical systems. At design time a digital twin can in fact be used to represent the to-be system reflecting its characteristics in the digital world and especially to conduct simulations before the system is actually implemented.

This paper reports about an approach for the design and implementation of a Digital Twin Prototype for a project involving an IoT life-saving system designed to support the rescue operation of people during a seismic event. The approach as well as the software tool can be adopted to other IoT or Cyber-Physical systems.

Keywords: Digital Twin, Digital Twin Prototype, Internet of Things, 3D Modelling, 3D Simulation

1 Introduction & Motivation

There has been a rapid rise of interest in the potential of Digital Twins (DTs) to transform a vast range of Internet of Things (IoT) and Cyber-Physical System (CPS) applications [15]. The field of DT is appearing to undergo a large increase in attention from both industry and academia. The 2023 Gartner emerging tech impact radar, places DTs among the most impactful emerging technologies and trends [22]. In addition, according to a 2022 report, nearly 60% of executives across a broad spectrum of industry plan to incorporate DTs within their operations by 2028 [19].

In academia, an increasing amount of research papers is being published every year. We can notice works ranging from those that investigate the definition of DT [3, 8, 9, 20], to more extensive works that cover several aspects of the DT topic such as modelling and enabling technologies [6, 17, 21], to works that focus on DT applications to specific domains [7].

In this research work we explore the implementation of a DT solution for the SAFE scenario. “S.A.F.E. - Sustainable design of Antiseismic Furniture as smart life-saving systems during an Earthquake” was an Industrial Research project[1] concluded in 2021, that aimed to design and implement smart and life-saving furniture systems in case of earthquake for school and office contexts [18]. A deployment of the SAFE “system” to an actual classroom of a school in the Marche Region of Italy is planned as part of another project called VITALITY[2].

The design and implementation in a real environment of the SAFE system is complex both in terms of components to consider (furniture, IoT sensors, ICT infrastructure), and as regards to the validation of their integrated operations. Testing operations of the entire system are particularly challenging since they require the entire system to be deployed or a small-scale physical prototype to be created facing the challenge of trying to replicate the conditions of an earthquake. It therefore becomes of extreme importance to be able to anticipate the validation of the system right from the design stages, making evaluations and behavior simulations even before the components themselves are actually installed. The definition of a DT in order to study the system before installing it in the physical environment, could bring several benefits to the SAFE scenario. Especially, we refer to the notion of Digital Twin Prototype (DTP) [9] since the corresponding physical twin of the SAFE scenario does not exists yet.

In this paper we report our experience in the design and development of a DTP for the SAFE scenario especially focusing on the process we adopted to graphically modeling and simulating the scenario. With respect to the implementation of DTs, IoT platforms are often seen as the starting point. According to [19] by 2028 the 90% of IoT platforms will be extended to support DTs. To implement our SAFE DTP we mainly relied on the ThingsBoard[3] IoT platform which we extended to support 3D modeling and visualization of a scenario, as well as 3D simulation of a scene in which multiple devices are deployed. The ThingsBoard extension is avaialbe at https://pros.unicam.it/digitaltwin/dtplatform. The interested reader can take inspiration from our approach as well as make use of our tool to start implementing a DTP of his own scenario.

The rest of the paper is structured as following. In Section 2 we report details about the SAFE project. Considering the complexity of the SAFE scenario, we focus on the PIR-based motion detection device (we call it SAFE PIR) of which we report a description of its dynamic behavior. We then discuss in Section 3 the process we followed to design a DTP of a SAFE classroom based on a virtual deployment of multiple SAFE PIRs. In Section 4 we report about the modelling of the SAFE scenario while in Section 5 we report about a mechanism we defined for simulating the SAFE scenario within ThingsBoard. We report in Section 6 a discussion on functionalities that our DTP enables as well as some limitations. Section 7 reports about related work that focus on the implementation of DTs for complex scenarios. We close the paper with Section 8 by drawing conclusions.

---

[1] SAFE project: http://projects.cs.unicam.it/safeproject/index.html

[2] VITALITY project: https://vitality-spoke6.unicam.it/en/

[3] ThingsBoard IoT platform: https://thingsboard.io/

2 The SAFE Project

In this section we first provide an overview of the SAFE project for then focusing on the SAFE PIR device and its dynamic behavior.

2.1 Project Overview

The main objective of the SAFE project was the design and prototyping of furniture for schools and offices capable of transforming themselves into intelligent systems of passive and "life-saving" protection of people during an earthquake, integrating technical-scientific knowledge and skills as those of Industrial Design, Structural Engineering, Computer Science and Chemistry and facilitating a process of cross-fertilization. The basic idea of the project resulted from the observation of a recurring phenomenon: during an earthquake, furniture and mobile equipment become obstacles that aggravate the dangerous conditions or, on the contrary, represent a casual protection of life in the event of collapses.

The challenge of the project was to innovate the design of traditional furniture (e.g. desk, equipped wall, etc.), for schools and offices, transforming them into intelligent systems through the integration of IoT sensors (SAFE devices) and a related ICT infrastructure. The ICT infrastructure was in charge of integrating the SAFE devices data through local gateways and a dedicated instance of the ThingsBoard IoT remote platform used to provide the basic monitoring and management services. Data collected from the SAFE devices could then be used to support the localization and rescue of survivors under the rubble during an earthquake [18].

The SAFE devices consisted of battery powered wireless IoT devices, designed to be integrated in the SAFE furniture as shown in Figure 1 and Figure 2. The primary objective of these devices, in case of a earthquake, is to detect and communicate whether there are persons being protected by the SAFE furniture. The information then is made available to rescue teams supporting localization and rescue activities. Given the importance of detecting people under the smart furniture, the SAFE PIR device has been developed in such a way to fit within the furniture and to adapt its behavior in case of a seismic event.

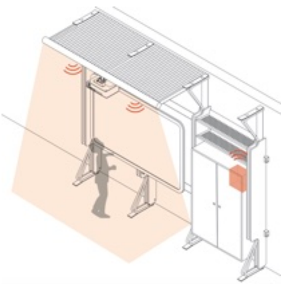


Fig. 1. SAFE Equipped Wall.

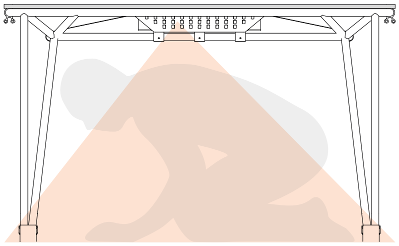


Fig. 2. SAFE Desk.

2.2 SAFE PIR Behavior

The SAFE PIR implements a *dynamic behavior* that changes from *Peace* Mode - the modality adopted when no seismic event is present - to *War* Mode - the modality adopted when a seismic event occurs. We describe and illustrate such a behavior by means of two BPMN models. Fig. 3 reports the behavior in *Peace* Mode (i.e., before the earthquake) and Fig. 4 reports the behavior in *War* Mode (i.e., during and after the earthquake). Considering that the BPMN notation lately acquired relevance in the modelling of IoT and CPS systems [1, 2, 24], it came natural for us to conceptualize the PIR behavior using such a notation. The use of BPMN gives the advantage of using a notation that is easily understandable, even to non-expert users.

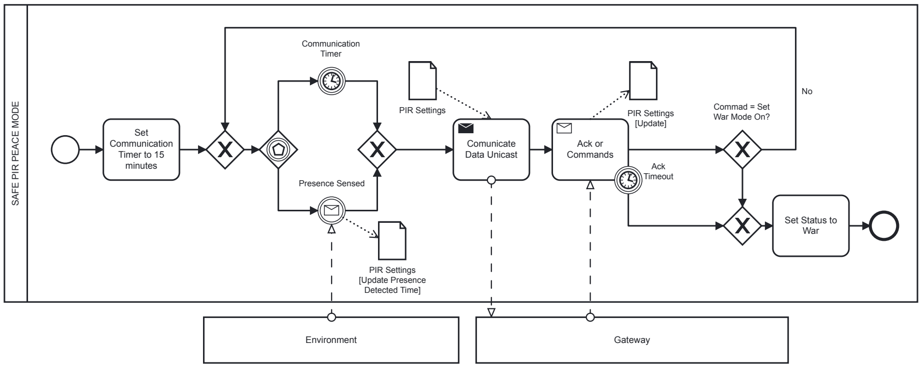


Fig. 3. PEACE Mode Behavior of the SAFE PIR represented with the BPMN notation.

Peace Mode behavior. The default SAFE PIR's behavior is the one we indicate with *Peace Mode*. The first activity performed in Peace Mode sets the communication timer to 30 minutes. This timer will be used for sending regular *keep-alive messages* to ensure the communication between sensors and gateway is active, as well as for sending diagnostic information about the device (i.e., *battery status*, *device-temperature*, etc.). Next, either the *Communication Timer* expires (i.e., 30 minutes have passed) or a movement is detected through the arrival of a *sensing* message from the *Environment* that is represented in the model as a black box pool. The SAFE PIR sensor is triggered whenever a temperature variation is detected within its field of view. This variation can be associated with a movement of any heat-emitting object, such as people or animals.

After one of the two events occurs, the SAFE PIR activates a Unicast Communication (*Communicate Data Unicast*) for sending a message to the *Gateway*, represented in the model as a black box pool. Then, the SAFE PIR waits to receive an *Ack* message that could also include some *Commands* used to request to switch in War Mode or to set different values for Ack and communication timers. Here, three situation may occur. 1) An Ack message is received and no command has been provided. 2) An *Ack Timeout* occurs while waiting for the Ack message, this means the gateway for some reason is not reachable anymore and the *War* Mode is activated. 3) An Ack message from the gateway is received and it contains some commands the SAFE PIR will have to consider, for instance a command requesting the SAFE PIR to switch to War mode due to an earthquake.[4] After handling any of the mentioned cases, the SAFE PIR can either exit from Peace Mode or go back to wait for the Communication Timer to expire or for a presence to be sensed.

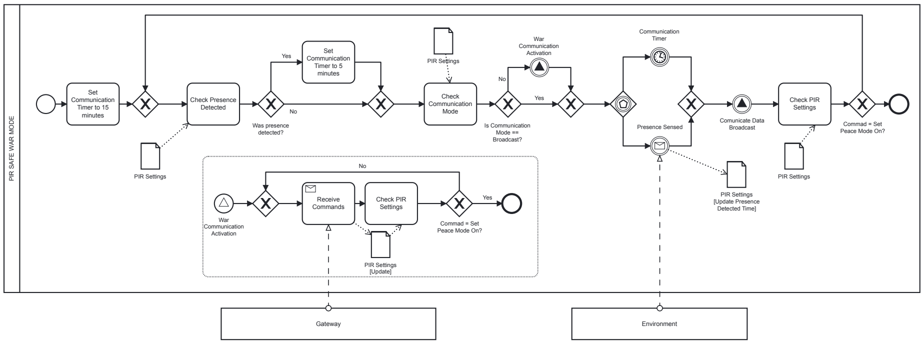


Fig. 4. WAR Mode behavior of the SAFE PIR represented with the BPMN notation.

In *War Mode*, the *Check Presence Detected* activity checks if a movement has been detected (i.e., at least one *sensing* message has been received). If it is the case, the SAFE PIR updates the communication timer (e.g., to 1 minute) increasing therefore the rate of messages sent so to generate more accurate information for the rescue teams.

Then, the SAFE PIR activates, if not already activated, a broadcast communication modality. This means the transmission system is turned on to continuously listen for possible incoming commands from a gateway. In case the local gateway is not reachable anymore, a possible flying gateway (by means of a drone) could be deployed and could act as gateway for collecting sensed information from the various SAFE PIR sensors and also for requesting an update of the communication timer or a possible switch back to Peace Mode. We represented this behavior through the *War Communication Activation* signal that triggers the corresponding event subprocess.

Finally, the SAFE PIR waits for either the communication timer to expire (e.g., after 1 minute) or for sensing movements from the person protected by the SAFE furniture. In both cases, a new message will be broadcasted. Then the device checks whether or not to maintain the War Mode behavior.

---

[4] The SAFE gateway implements an Earthquake Early-Warning detection system through a specific accelerometer. In case of an earthquake, the gateway sends a command to the SAFE devices requesting them to switch to War Mode.

3 The Adopted Process for a SAFE DTP

Among all the characteristics that a DT can have [3, 13], for the SAFE scenario we focused on 3D modeling and 3D simulation. We reported in Fig. 5 the process we adopted to design and implement the SAFE DTP.

As first step, we designed the 3D model of the SAFE PIR reflecting the real device. Then we associated the 3D model to digital representations registered in ThingsBoard, we refer to them as *Digital Devices*. The digital devices can be enriched with attributes, treated as key-value pairs, to describe characteristics about the physical devices such as: *name*, *description*, *firmware version*, *latitude*, *longitude*, etc. In addition, within an IoT platform like ThingsBoard, telemetry data coming from the physical devices are associated to the digital ones and made available for inspection so to allow monitoring the actual state of the physical device and of the environment's aspects it perceives. Then we designed the 3D model of a real classroom and we combined it with the SAFE PIR digital devices and related 3D models, to design a 3D scene of the SAFE scenario.

After designing the 3D SAFE scenario, we focused on the steps needed to simulate it. As first, we encoded the SAFE PIR behavior in ThingsBoard as described in Sec. 2.2. Then, we designed and executed the SAFE simulation using the 3D scene we previously defined with the objective to test the SAFE PIR behavior.

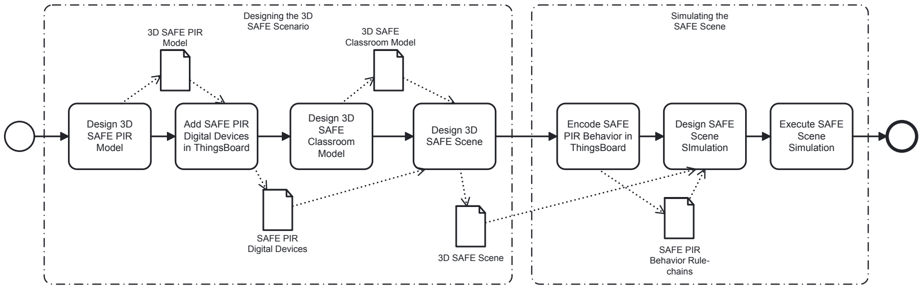


Fig. 5. Steps for implementing the SAFE Digital Twin Prototype.

In the next sections we describe in detail how we conducted the modelling and simulation of the SAFE scenario.

4 Modelling the SAFE Scenario

For designing the 3D model of the SAFE PIR device and of the SAFE classroom we used Blender[5] a free and open-source 3D creation suite. We started our modelling activities from a real SAFE PIR device reported in Fig. 6. The SAFE PIR is composed of: a printed circuit board (PCB) with the PIR sensor, a battery pack, an antenna, and the wires that connects them. As it can be seen from Fig. 7 we faithfully designed the 3D model of the SAFE PIR device and its components.

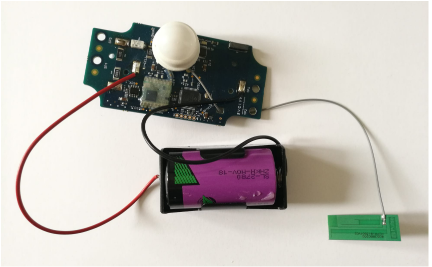


Fig. 6. Real SAFE PIR device.

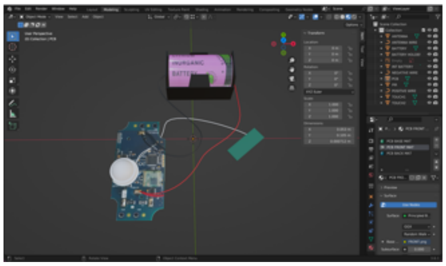
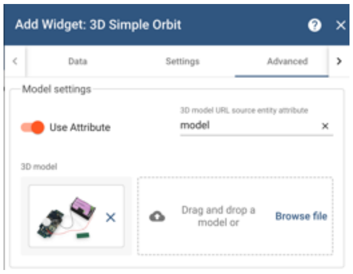


Fig. 7. SAFE PIR 3D Model in Blender.

The SAFE PIR, like any other IoT device, can be registered on the ThingsBoard platform leading to the definition of a digital device. To associate the 3D model to the SAFE PIR digital device we developed a widget, partially shown in Fig. 8. The widget offers a straightforward and intuitive 3D visualization of a single object. It allows users to rotate, zoom in/out, and visualize the exploded view of the object. With the simple orbit controls, users can easily manipulate the object's orientation and gain a comprehensive understanding of its spatial features as shown in Fig. 9.



Add Widget: 3D Simple Orbit

Model settings

Use Attribute

3D model URL source entity attribute: model

3D model

Fig. 8. Adding 3D Model in ThingsBoard.

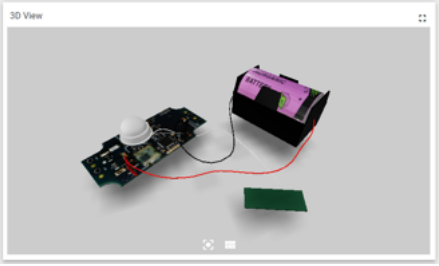


Fig. 9. 3D SAFE PIR in ThingsBoard.

---

[5] Blender https://www.blender.org/

After taking care of the SAFE PIR representation, we started to model the SAFE classroom in Fig. 10 with the various components such as walls, windows, doors, desks, chairs, etc. Again, we used Blender to design the 3D model of the SAFE classroom as reported in Fig. 11.

Fig. 10. Real SAFE classroom.

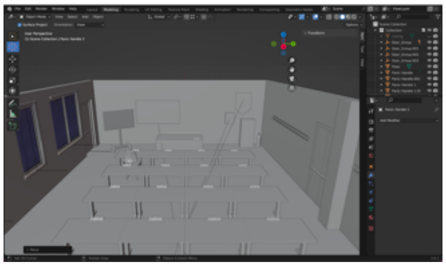


Fig. 11. 3D SAFE Classroom in Blender.

Since the SAFE classroom is not a single IoT device, we do not associate it with a digital device in ThingsBoard. Instead, by means of a widget that we developed, we designed the 3D scene to simulate by importing the 3D model of the environment and then incorporating the 3D models of the digital devices. Specifically, we designed the SAFE 3D scene modifying the 3D classroom model by adding the 3D SAFE PIR model, as can be seen in Fig. 12, and we adjusted the scene positioning the devices under the furniture, as to reflect what will be the real scenario, see Fig. 13.

Fig. 12. SAFE Scene Design.

Fig. 13. SAFE Pir device 3D model.

5 Simulating the SAFE Scenario

In this section we describe how we encoded the SAFE PIR behavior in ThingsBoard as wells as the mechanism we proposed to simulate the SAFE scenario.

5.1 Implementing the SAFE PIR Behavior

As anticipated in Sec 3, we encoded the SAFE PIR behavior described in Sec. 2.2 into ThingsBoard. We used the ThingsBoard *Rule Chain Editor*, that makes use of a low-code approach, as many other IoT platforms do [12], to allow users to define complex rules in terms of connected control flows where certain conditions can trigger specific actions based on the data received from devices. In Fig. 14 we report, for presentation purpose, an excerpt of the rule-chains we defined. Especially, part a) of the figure reports the root rule-chain that combines three other rule-chains: *Set CTimer & CMode*, *CTimer or Presence*, and *Communication*. Part b) of the figure reports the expanded *Set CTimer & CMode* rule-chain. We report these rule-chains as examples to illustrate how we used Thingsboard to encode the SAFE PIR's behavior.

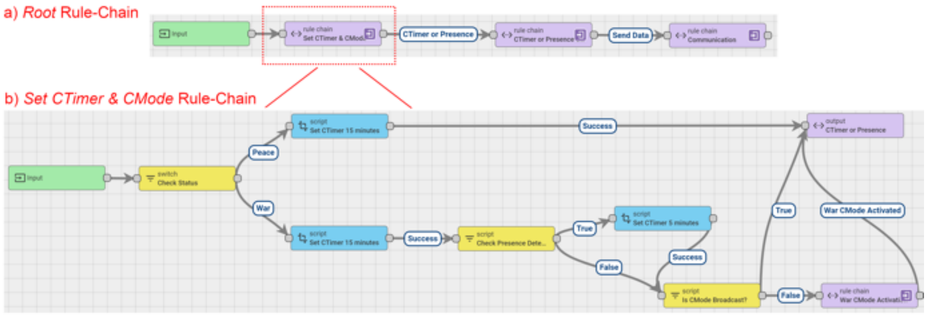


Fig. 14. SAFE PIR's behavior encoded via Thingsboard's rule-chains.

The root rule-chain starts by invoking the *Set CTimer & CMode* rule-chain that checks whether the SAFE PIR is set in Peace or in War mode. Then, it sets the communication timer to 30 minutes. But, if the PIR is in War mode, it also checks whether a presence is detected. In case a presence is detected then it sets the communication timer to 1 minute and, in case the broadcast communication is not already active, it switches to War communication mode, i.e., broadcast, (handled by the *War CMode Activate* rule-chain). At the end of the rule-chain, the control moves to the next *CTimer or Presence* rule-chain that takes care of handling the possible upcoming events (i.e., the communication timer expiration or the sensing of a presence). The last rule-chain *Communication*, takes care of handling the Communication with a Gateway.

5.2 Simulation Mechanism

Every rule-chain defined in ThingsBoard can be executed by means of a *Rule Chain Engine*, a powerful tool mainly used for processing and analyzing data generated by IoT devices. We propose a simulation mechanism that leverages the capability of the engine to actually simulate the SAFE PIR's behavior.

Besides the concept of digital entity that we use to represent the SAFE PIR device in ThingsBoard, we introduced the concept of *Simulated Device* (SIM-PIR in Fig. 15). In our case, simulated devices are basically copies of SAFE PIR digital devices and inherit all their characteristics as well as possible associated rule-chains. The simulated devices are the ones actually used for running the simulations. We made this distinction for avoiding simulated telemetry data to override real telemetries coming from the physical world and reflected on the SAFE PIR digital devices.

Once the rule-chain and the simulated devices have been defined, we can play the role of a *Simulated Designer* and create and simulate, by means of a *simulation widget* we developed, the behavior of IoT or CPS systems in a virtual environment. For doing so we need some programming skills, especially some familiarity with the Three.js[6] and the cannon-es[7] libraries is required to adapt the *3D Scene* adding additional 3D objects and for handling the physics.

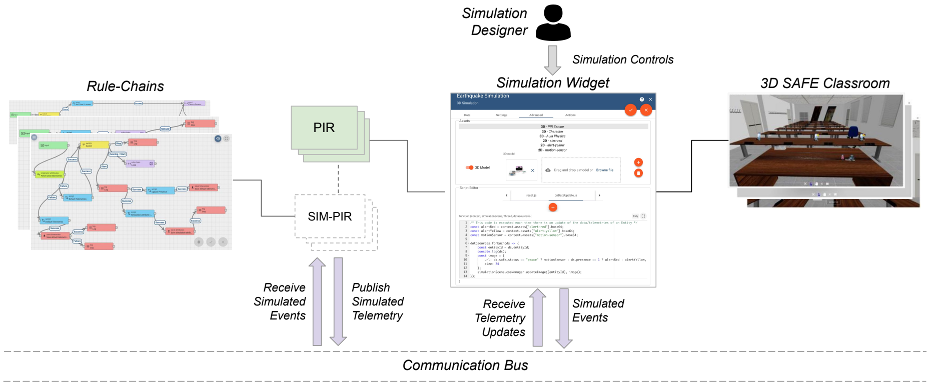


Fig. 15. A representation of the components involved in the design of a 3D simulation in ThingsBoard.

In Fig. 15 we illustrate the 3D simulation widget with the various components involved in the design and execution of a 3D simulation. The components communicate by means of a *Communication Bus* that abstracts the communication layer of ThingsBoard. Especially, when a simulation is activated by a user, simulated events in the simulated environment may occur. Such simulated

---

[6] Three.js: https://threejs.org/
[7] Cannon-es: https://pmndrs.github.io/cannon-es/

events are published on the communication bus and received by the corresponding simulated digital device which handles the event by updating its 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 virtual environment.

5.3 SAFE Simulation

We used the simulation mechanism previously described to design and execute a *3D Simulation* of the SAFE scenario. In Fig. 16 we show the initial setup of the simulation which includes the SAFE classroom with three SAFE PIR devices and two humanoids that simulate the presence of two persons in the room. The simulation has been programmed in such a way to simulate an earthquake scenario. As soon as a user starts the simulation, the classroom will start shaking, the objects will be affected by the forces applied by the earthquake and the individuals within the scene will seek out the nearest smart furniture equipped with the SAFE PIR device to take shelter under it.

A 2D icon is associated to each SAFE PIR in the 3D scene. The icon will change reflecting the PIR's behavior change from *Peace mode* to *War mode*. A white icon is used to indicate that the SAFE PIR is in peace mode, as in Fig. 16. A yellow triangle represents that the SAFE PIR is in war mode but that no presence has been detected, while a red triangle represents that the SAFE PIR is in war mode and a presence has been detected, both are shown in Fig. 17.

The detection of a presence is simulated by the collision occurring between the cone collider, that represents the SAFE PIR's coverage, and the humanoid collider. When a presence is detected, an event is published on the communication bus and the corresponding SIM-PIR will receive it and update its telemetry. In turn, this update triggers the rule-chain that predicates over that telemetry.

Fig. 16. Initial 3D simulation settings.

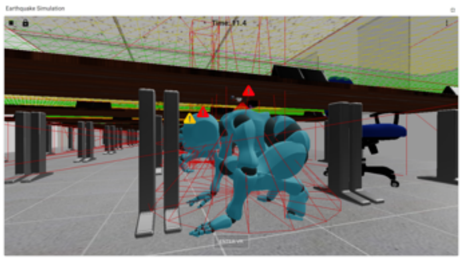


Fig. 17. Earthquake simulation.

6 Discussion

In this section we discuss the practical implications of the proposed SAFE DTP implemented in ThingsBoard, thanks to the extensions we developed, and we report about some limitations of the present solution.

6.1 Enabled Functionalities

The use of a DTP for the SAFE scenario enables the possibility to test the behavior of the SAFE PIR devices before actually deploying them in a real setting. By means of a platform that allows to define digital devices with attributes to reflect the real characteristics of the physical device (dimensions, components disposition, device coverage, etc.) we can use *3D modelling* to effectively describe a real-world environment and reflect such characteristics. For instance we can use the modelling to display the device coverage, see Fig. 18, and plan an optimal dispositions of the furniture to avoid device interference.

While 3D modeling allows us to faithfully represent a real scenario, *3D simulation* enables us to evaluate the run time behavior under different hypothesis. The possibility to setup 3D simulations of the SAFE classroom, allows us to test the SAFE PIR behavior at design phase, according to the desired simulated conditions. This allows us to asses whether the devices behave correctly according to the simulated condition, and before actually deploying it in the real scenario.

Moreover, when the physical SAFE devices will be deployed in the physical classroom we will be able to link digital and physical devices. This will allow, thanks to the extensions we developed, to visualize the actual data coming from the physical twin, directly within the 3D model as shown in Fig. 19, enabling therefore the possibility to conduct *3D Monitoring* of the physical environment. We will also be able to start simulations from real telemetry data.

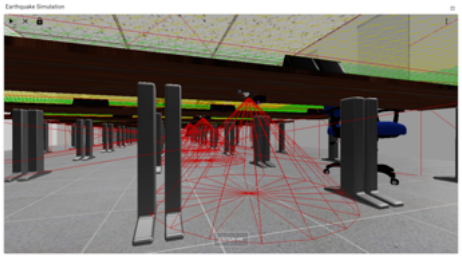


Fig. 18. Visualized PIR's coverage.

Fig. 19. SAFE Pir device 3D model.

6.2 Limitations

We recognize and report in the following limitations of the presented approach.

The design of 3D models especially for non expert users may be a cumbersome activity. In several domain such as manufacturing, construction, etc., digital 3D models are already being used therefore as a possible future direction we envision the possibility to integrate the support for such models (STL, IFC, etc.) directly into the platform used for developing the DT solution.

The encoding of devices behavior by means of rule-chains may not scale well when the behavior to represent is complex, in fact the behavior we represented with two BPMN models required thirteen ThingsBoard rule-chains that we manually encoded. More complex scenarios may require the manual design of too many rule-chains. For solving this issue we envision the possibility to define a parser from BPMN models to ThingsBoard rule-chains or to directly combine a BPMN engine with ThingsBoard.

Designing a graphical simulation with our ThingsBoard extension requires some programming skills, therefore we envision the possibility to define alternative approaches to facilitate this step, i.e., a model drive approach could be defined to support this step.

At the present stage, it is possible to visualize the execution of the simulation only by means of the 3D scene and by looking at the ThingsBoard log. However it would be interesting to be able to check the graphical rule-chain and see from there which is the action the PIR is performing at a specific moment.

7 Related Work

Several research work focus on the design and development of DTs using different approaches dependent on the kind of scenario and requirements needed to be fulfilled. We report in the following a non-exhaustive list of related work that focus on implementing DTs solutions for complex scenarios.

In [11] the authors use DTs and a related IoT platform to address congestion problem caused by container trucks in port areas scenario. The decision making support system implements a Python simulation framework aided with advanced visualization modules. The behavior to simulate is conceptualized by means of BPMN models and then parsed into python modules executed by a simulator developed using Python SimPy.

In [10] the authors propose an interactive DT platform based on Unity3D to implement the simulation and visualization features for offshore wind farms tracking conditions. The simulation layer relies on Functional Mock-up Unit (FMU) and Matlab to model the wind turbine and imported though Unity FMI Add-on.

In [14] the authors propose an hospital DT model based on discrete event simulation and IoT computing devices to optimise health care services. The simulation model relies on FlexSim HealthCare as 3D simulation and modelling tool used to evaluate and visualize patients and staff flows scenarios within the simulated model run-cycles.

In [4] the authors report an approach for supporting the representation, simulation, and visualization of digital process twins of autonomous systems. The approach has been built on top of BPMN collaborations, for representing the system behavior, the MIDA tool, for simulating the system, and Gazebo for visualizing the outcomes. A demonstration scenario is implemented regarding an autonomous system for airport luggage handling.

In [5] the authors presents a novel concept of executable digital process twins to effectively enable the monitoring, analysis, and refinement of process-driven systems. They illustrate how to implement an executable digital process twin in a cooperative multi-robot scenario. The approach is supported by a tool PROWIN to implement the monitoring of the executed system from the process and the physical perspectives. The tool also allows the deployment of a refined process model into the robots, thus enabling the synchronization between the physical and the digital systems. They assess the approach by means of a BPMN-driven multi-robot system deployed in a warehouse.

In [23] a microservice architecture to support the implementation of DTs for IoT-Enhanced Business Processes is presented. This architectural solution is supported by a model-driven development approach, that allows to move from modelling to implementation of the DT for the IoT-Enhanced Business Process. A scenario concerning a CO2 Management system for a smart library is reported and used as a demonstrator.

The related work previously reported mostly rely on a composition of tools for supporting the development of DTs. This means that users need to install and configure all these tools to make them work together, which can be time-consuming and complex. Furthermore, from a developer's perspective, this requires knowledge of all the tools used and an understanding on how to extend the composition for further improvements. Differently from them, in our work we mainly focus on the usage of a single IoT platform extended to support DT aspects. In addition some works tend to be too specific for their use case, limiting their usability in different contexts. This can make it challenging to adapt the solution to different use cases or scenarios.

With respect to the implementation of DTs solutions, DT platforms have started to appear in the market such as: *Azure Digital Twins*, *AWS IoT TwinMaker*, *iTwin Bentley*, *Ansys Twin Builder*, and many others. Most of those platforms are proprietary and have different characteristics and provide different supports for DTs [16]. In our case we chose to develop a DT solution extending the ThingsBoard open source IoT platform instead of using a proprietary DT platform, remaining also consistent to the ICT infrastructure designed in the SAFE project which already relied on the ThingsBoard IoT platform.

8 Conclusion

In this work, we presented the SAFE scenario and described the process and tools we adopted to implement its Digital Twin Prototype. Once the real devices will be deployed in the actual environment, we will be able to perform 3D monitoring and to run up-front simulations starting from actual telemetries of the physical devices. This can enable various analysis of the IoT or CPS system deployed as well as possible predictive maintenance operations. We also discussed some limitations of the presented approach and proposed ways to overcome them as possible future work. The approach as well as the software tool can be adopted to implement Digital Twins for other IoT or CPS scenarios.

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. Bourr, K., Corradini, F., Pettinari, S., Re, B., Rossi, L., Tiezzi, F.: Disciplined use of BPMN for mission modeling of multi-robot systems. In: Proceedings of the Forum at Practice of Enterprise Modeling, Riga, Latvia, November 24-26, 2021. CEUR Workshop Proceedings, vol. 3045, pp. 1–10 (2021)
2. Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: A systematic literature review on iot-aware business process modeling views, requirements and notations. Softw. Syst. Model. **22**(3), 969–1004 (2023)
3. 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)
4. Corradini, F., Pettinari, S., Re, B., Rossi, L., Tiezzi, F.: An approach to support digital process twin. In: IEEE DASC/PiCom/CBDCom/CyberSciTech 2022, Falerna, Italy, September 12-15, 2022. pp. 1–4. IEEE (2022)
5. Corradini, F., Pettinari, S., Re, B., Rossi, L., Tiezzi, F.: Executable digital process twins: Towards the enhancement of process-driven systems. Big Data and Cognitive Computing **7**(3) (2023)
6. Dalibor, M., Jansen, N., Rumpe, B., Schmalzing, D., Wachtmeister, L., Wimmer, M., Wortmann, A.: A cross-domain systematic mapping study on software engineering for digital twins. J. Syst. Softw. **193**, 111361 (2022)
7. Fuller, A., Fan, Z., Day, C., Barlow, C.: Digital twin: Enabling technologies, challenges and open research. IEEE access **8**, 108952–108971 (2020)
8. Grieves, M.: Intelligent digital twins and the development and management of complex systems. Digital Twin **2**(8), 1–8 (2022)
9. 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)

10. Hasan, A., Hu, Z., Haghshenas, A., Karlsen, A., Alaliyat, S., Cali, U.: An interactive digital twin platform for offshore wind farms' development. In: Digital Twin Driven Intelligent Systems and Emerging Metaverse, pp. 269–281. Springer (2023)
11. Hofmann, W., Branding, F.: Implementation of an iot-and cloud-based digital twin for real-time decision support in port operations. IFAC-PapersOnLine **52**(13), 2104–2109 (2019)
12. Ihirwe, F., Ruscio, D.D., Mazzini, S., Pierini, P., Pierantonio, A.: Low-code engineering for internet of things: a state of research. In: ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems, Canada, 18-23 October, 2020, Companion Proceedings. pp. 74:1–74:8. ACM (2020)
13. Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the digital twin: A systematic literature review. CIRP journal of manufacturing science and technology **29**, 36–52 (2020)
14. Karakra, A., Fontanili, F., Lamine, E., Lamothe, J., Taweel, A.: Pervasive Computing Integrated Discrete Event Simulation for a Hospital Digital Twin. In: 15th IEEE/ACS International Conference on Computer Systems and Applications, Aqaba, Jordan, October 28 - Nov. 1, 2018. pp. 1–6. IEEE Computer Society (2018)
15. Larsen, P.G., Fitzgerald, J., Woodcock, J.: How do we engineer trustworthy digital twins? Research Directions: Cyber-Physical Systems p. 1–6 (2023)
16. 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)
17. Mihai, S., Yaqoob, M., Hung, D.V., Davis, W., Towakel, P., Raza, M., Karamanoglu, M., Barn, B., Shetve, D., Prasad, R.V., Venkataraman, H., Trestian, R., Nguyen, H.X.: Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. IEEE Commun. Surv. & Tutorials **24**(4), 2255–2291 (2022)
18. Pietroni, L., Mascitti, J., Galloppo, D.: Life-saving furniture during an earthquake. intelligent, interconnected and interacting. AGATHÓN— International Journal of Architecture, Art and Design **10**, 218–229 (2021)
19. Researchandmarkets: Digital twins market by technology, twinning type, cyber to-physical solutions, use cases and applications in industry verticals 2022–2027. https://www.researchandmarkets.com/report/digital-twin
20. Semeraro, C., Lezoche, M., Panetto, H., Dassisti, M.: Digital twin paradigm: A systematic literature review. Comput. Ind. **130**, 103469 (2021)
21. Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B.D., Todd, M.D., Mahadevan, S., Hu, C., Hu, Z.: A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies. Structural and Multidisciplinary Optimization **65**(12), 354 (2022)
22. Tuong, N., Jump, A., Casey, D.: Emerging Tech Impact Radar: 2023: Gartner Research Excerpt. https://www.gartner.com/en/doc/emerging-technologies-and-trends-impact-radar-excerpt
23. Valderas, P.: Supporting the implementation of digital twins for iot-enhanced bps. In: International Conference on Research Challenges in Information Science. pp. 222–238. Springer (2023)
24. Valderas, P., Torres, V., Serral, E.: Towards an interdisciplinary development of iot-enhanced business processes. Bus. Inf. Syst. Eng. **65**(1), 25–48 (2023)