Digital twin for IoT environments : a testing and simulation tool

Luong Nguyen, Mariana Segovia, Wissam Mallouli, Edgardo Montes de Oca, Ana R Cavalli

Digital Twin for IoT Environments : a Testing and Simulation Tool

Luong Nguyen[1], Mariana Segovia[2], Wissam Mallouli[1], Edgardo Montes de Oca[1], and Ana R. Cavalli[1,2]

[1] Montimage, 39 rue Bobillot, 75013 Paris, France
{luong.nguyen, wissam.mallouli, edgardo.montesdeoca}@montimage.com
[2] Telecom SudParis, 9 rue Charles Fourier, 91011 Evry, France
{ana.cavalli, segovia}@telecom-sudparis.eu

Abstract. Digital Twin (DT) is one of the pillars of modern information technologies that plays an important role on industry's digitalization. A DT is composed of a real physical object, a virtual abstraction of the object and a bidirectional data flow between the physical and virtual components. This paper presents a DT-based tool, called TaS, to easily test and simulate IoT environments. The objective is to improve the testing methodologies in IoT systems to evaluate the possible impact of it on the physical world. We provide the conditions to test, predict errors and stress application depending on hardware, software and real world physical process. The tool is based on the DT concept in order to detect and predict failures in evolving IoT environments. In particular, the way to prepare the DT to support fault injection and cybersecurity threats is analyzed. The TaS tool is tested through an industrial case study, the Intelligent Transport System (ITS) provided by the INDRA company. Results of experiments are presented that show that our DT is closely linked to the real world.

Keywords: Digital Twins, IoT, Sensors, Actuators, Gateway, Simulation,Testing

1 Introduction

Testing is a crucial step of any software development process [3]. As a result, various test cases (e.g., unit tests, integration tests, regression tests, system tests) need to be designed and executed in a production-like environment that reproduces the same conditions where the software under test would run. However, having access to such an environment may be hard to achieve and it is even particularly challenging in the IoT area.
The access to IoT devices might be non-trivial or limited due to many factors. For example, networks of physically deployed devices are typically devoted to production software. Testing applications on top of those networks might involve additional testing software, which might affect the overall performance and the revenue generated by the devices (e.g., applications need to be stopped to load their new versions).
Software simulators proved to be valuable in easing the verification of the software requirements. They provide software developers a testing environment to at least manage the execution of test cases. IoT Testbeds play a similar role in testing IoT applications. They offer a deployed network of IoT devices where developers

can upload their applications and test their software in a physical environment. IoT-Lab [1] and SmartSantander [17] are good examples of IoT testbeds. Testbeds often have a predefined fixed-configuration and architecture. They are also usually shared with other users, which can be a problem for measuring application quality. Hence, this problem might make simulators more attractive since they provide a more customized and controlled environment. Furthermore, simulators avoid the need for a more expensive physical network of devices.
The main issue regarding simulators is that they are not directly linked to the real environment and any evolution of this latter (e.g., addition or deletion of a new IoT device or gateway) is not automatically taken into account in the simulation mode. Also, physical process dynamics may be hard to be reproduced in simulations. As a result, physical properties and events, such as process disturbances or devices failure, may not be quantified during the software testing process. Besides, simulation can rely on predefined scenarios that can have different behaviours in real environments since simulation is based on the abstraction of some layers. The continuous monitoring of real systems is needed to feed simulators in order to have more accurate results. In addition, recommendations from simulators can be taken into account in the real world if a bidirectional relationship between these two worlds exist. This is exactly the essence of Digital Twins.
The main contribution of our paper is the design of a tool, called TaS (stands for Test and Simulation), based on the concept of Digital Twin, to simulate, test and predict errors in real IoT systems. The tool supports functional and non-functional testing through the real-time connection of the physical system to a new software version deployed in the DT. This way, it is possible to verify that the changes made in the code do not impact the existing software functionality. Also, the DT may be used to elaborate a what-if analysis resulting in a better evaluation of attacks, error cases, scalability and performance stress situations. For example, it is possible to perturb the system to test unexpected scenarios and analyze the response. TaS has been validated through different experiments performed in the context of H2020 ENACT project[3].
The paper is organized as follows: Section 2 presents several solutions for the simulation of IoT environments as well as the usage of DT for this kind of technology. Section 3 presents the basics to understand the concepts of DT as well as simulation and testing. Section 4, presents the TaS tool, its architecture and different details of its implementation. In section 5, we present the application of such DT-based Test and Simulation tool on an industrial experimental case study called ITS. Finally, we conclude the paper and discuss future work in section 6.

2 Related Work

In recent years, both academia and the commercial market offered solutions in the design of DT. Following we present some relevant works regarding DT as well as simulation and testing for IoT systems. We also explain the existing challenges in IoT applications and how our approach can help to solve these limitations.

Digital Twins — are a digital representation of a physical object or system or a system of systems (like an IoT network). The technology behind Digital Twins

[3] https://www.enact-project.eu/

has expanded to include complex elements such as buildings, factories and networks, and some even consider that people and processes can have DTs.
A DT is composed of a virtual object that models a physical component. Both components exchange information and the virtual object continually adapts to operational changes based on the collected data from the physical component. The connection between the physical and virtual objects can forecast the future of the physical component using the collected data [19]. This way, DTs supply a system with information and operating status providing capabilities to create new business models and decision support systems. Also, it is possible to make more accurate predictions and information-based decisions using analytic, predictive diagnosis, and performance optimization. Other uses of DT include reducing costs and risks, improving efficiency, security and resilience.
The idea first arose at NASA, where full-scale mockups of early space capsules, used on the ground to mirror and diagnose problems in orbit, eventually gave way to fully digital simulations [14]. But the term became very popular when Gartner named DTs as one of its top 10 strategic technology trends for 2017 [4] saying that within three to five years, "billions of things will be represented by Digital Twins, a dynamic software model of a physical thing or system". In essence, a Digital Twin is a computer program that takes real-world data about a physical object or system as inputs and produces as outputs predictions or simulations of how that physical object or system will be affected by those inputs.

IoT Simulation and Testing — The field of simulation and testing in IoT also has gained momentum when it comes to generating novel, cutting-edge ideas. In the recent years, academia proposed several IoT simulators each mostly focusing on a particular layer of the communication stack. For instance, Cooja[5] and OMNeT++[6] focus on simulating networking aspects of the systems. Other simulators, like SimIOT [18] or IOTsim [22], focus on data analytics rather than lower aspects of the systems. Another approach, like iFogSim [7], try to perform a complete simulation. However, having a full stack simulation from a single component or product can be challenging. Other alternatives proposed hybrid models, such as [2], which try to leverage several simulators, each for a particular layer, to reproduce the behaviour of a system from a holistic perspective.
The DTs are a development of modelling and simulation technology. Traditional simulation methods are of limited capabilities in evaluating system performance. By integrating IoT technology, DTs are the breakthrough of the existing limitations on the modelling and analysis capabilities of simulation [11]. The major difference between a simulation and a DT is the data interconnection that allows to exchange information between the physical and the virtual object, i.e., a simulation predicts future states of a physical system based on a set of initial assumptions [21]. However, a DT tracks the current and past states of the physical component that is being used in operation and is being simulated within the virtual object. Often the computational models which are used to infer the current state of the physical objects are the same models which can be used in simulation to predict future states. The simulation models can provide additional decision-making information for optimizing future operations, forecasting degradation mechanisms, and predicting future failures.

[4] https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017
[5] https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja
[6] https://omnetpp.org/

Research-based simulators often ignore problems such as the lack of standardization, which poses a challenge when it comes to creating synergies and interoperation between different simulators. A research-based simulator can be volatile and can change its application interface rapidly. This volatility generates extra overhead since developers need to adapt their code to the new changes. In addition, simulators created by research are often not maintained or discontinued, i.e. bugs remain and new features or improvements are not made. Nevertheless, some simulators are open source, allowing contributions from the community to their development and maintenance.

Testing Challenges — The research work regarding testing point out that there is a need for a complete set of tests and simulation solutions for IoT. Systems should be tested based on different scenarios that involve the generation and use of high amounts of sensor and actuator data, which is not always practical to set up in a given IoT environment, but serves to stress the boundaries of the environment in order to detect potential problems. This is exactly what we propose in this paper by conceiving a DT-based on a simulation and testing tool. Notice that the concept of DT for IoT has been used the first time in 2016 [6] where the authors proposed first ideas to define DT for industrial IoT. Then this concept has been studied mainly from a research point of view in [15] to address, e.g., smart grids and smart factories. The proposed tool that we present in this paper is generic enough so that it can be applied to different sectors (e-health, transport, telecommunication, etc.) and tackle different test objectives such as security, scalability, energy consumption. In addition, most of the existing DT proposals are designed for optimization of the physical object, system security and resilience, real-time monitoring, prediction of future behavior or training for operator users. Less attention has been paid to DTs applications to overcome the mentioned simulation limitations and improve IoT testing methodologies. Some proposals that have addressed this problem are analyzed as follows.

The paper [13] presents a survey providing the DT original definition and addressing the relevant aspects that a DT should support. It illustrates the application of the DT concept in four application scenarios. One of them is of particular interest for us, this regarding DT for sensors. Following this paper, sensors can be represented by a logical object or several ones, which are associate to the physical entities. In this DT, it is required that logical objects should be strongly synchronized with the physical objects. The objects are continuously updated. We have the same requirements regarding the sensor DT we defined in this paper. In addition, we go beyond this approach by developing a tool that implements the proposed solution.

In [16], it is presented an IoT-based DT of a cyber-physical system that interacts with the control system to ensure its proper operation. The proposed DT is validated on a distributed control system. Security measures are also implemented based on cloud computing. This work has the advantage that the proposed DT can contribute to mitigate individual as well as coordinated attacks.

The work in [9] proposes a tool to validate models of legacy systems. Their objective is to test the models of an existing production system through simulation and then incorporate this validated model in a DT. In this case, the proposition is oriented to create the modelling of an existing system. In our proposal, we go further by proposing a tool that test the whole system considering also the physical interaction.

The authors of [8] designed an open-source toolkit composed by five open-source tools (Eclipse Hono [7], Eclipse Ditto [8] , Apache Kafka [9], Influx DB [10] and Grafana [11]) for each data processing layer of IoT and DT reference architectures. The toolkit is evaluated using a benchmark dataset. The architecture of the toolkit is more complex than the proposed for our tool. Some experimentation showed that Hono and Ditto platforms have some limitation on massive packet processing [10] which may be a serious limitation to scale IoT applications.
In [4], the authors propose a DT for testing properties and characteristics of the physical object, i.e., for physical experimentation. Their work is motivated by the limited possibilities to physically experiment with convoy belts and how time-consuming this activity is. DT present a solution to create an environment to test objects using models without carrying out it physically. Our work provides also testing functionalities but with a focus in the software that controls the physical process and which are the possible impacts of it in the physical world. In this paper, we explore the creation of a DT to improve the development process of the software that controls the physical system. For that, we present a testing tool to evaluate functional and stress tests.

3 A Test and Simulation (TaS) Tool based on Digital Twin for IoT environment

This section contains three subsections. In the first subsection, the architecture of the tool called TaS enabler is presented. The second section presents its functionalities. The third section describes its implementation.

3.1 The approach and architecture of the tool

In this subsection, we present the architecture of the TaS enabler, which is based on the concept of DTs [5]. Figure 1 illustrates the TaS enabler architecture.
On the left-hand side, we have the system in a real (production) environment. The communication between the sensors, actuators with the IoT component is typically done via a broker. The sensors capture and send the surrounding information (e.g., "temperature") to the IoT system. Based on input data, the IoT system reacts differently and sends actuation data to change the actuator settings (e.g., "change the heating level").
On the right-hand side of the figure, we have the Smart IoT System (SIS) in a test environment and the TaS enabler. The system under test is the SIS that needs to be tested. The TaS enabler simulates sensors and actuators. The topology on the left side is very similar to the topology on the right side. The only difference is the simulated sensors and actuators. The simulated actuators collect the actuation data sent from the IoT system. The simulated sensors play the same role as the physical sensors providing the data signal to the IoT components. However, they are much more valuable than a physical sensor in terms of testing in the following ways:

[7] http://www.eclipse.org/hono/
[8] https://www.eclipse.org/ditto/
[9] http://kafka.apache.org/
[10] https://www.influxdata.com/
[11] https://grafana.com/

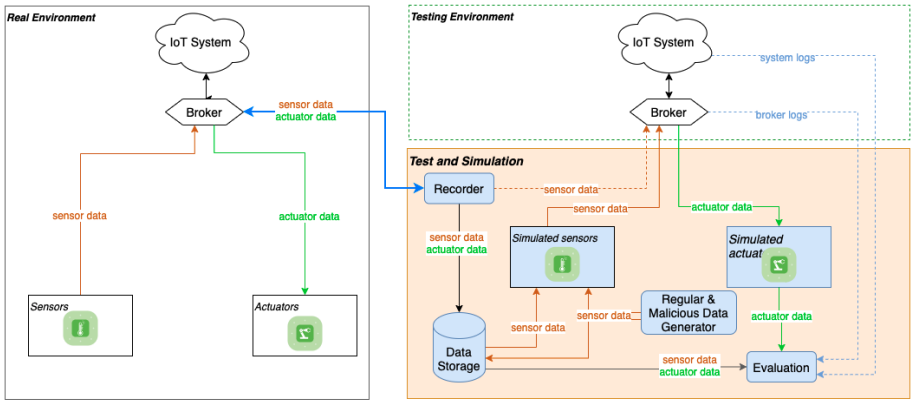

Fig. 1: Test and Simulation (TaS) Enabler approach and architecture

- Firstly, by using the dataset recorded from the physical environment, the simulated sensors can repeatedly simulate the surrounding environment at a specific time. In reality, an event may happen only once, but the simulated sensor can generate the same event as many times as needed for testing purposes.
- Secondly, the physical sensors passively capture the state of the surrounding environment. It can be challenging to obtain different data from the physical sensors. In contrast, the simulated sensors use the dataset in the Data Storage as a data source. Therefore, we can generate various testing scenarios by modifying the events in the Data Storage.
- Moreover, the TaS enabler also provides a module to manipulate the data from the sensors. The Regular and Malicious Data Generator can generate regular data to test the functionalities, operations, performance, and scalability (relying on pre-recorded data). It can also generate malicious data to test the resiliency of the system to attacks.

Besides the simulated sensors and actuators, the TaS enabler also provides some modules which support the testing process 1. The *Data Recorder* module records all the messages going through the broker in the physical environment. Each message can be considered as an event happening in the physical environment. Then, the recorded messages are forwarded to the broker in the testing environment. In this way, we have a "twin version" of the physical environment. What has happened in the physical environment is reproduced in the testing environment. Besides, the recorded messages are stored in a *Data Storage* as a dataset for later testing. The recorded dataset can be modified (muted) to create a new dataset, e.g., "change the event order", "delete an event", "add a new event". All the testing datasets are stored in the Data Storage. The *Regular and Malicious Data Generator* enables the simulation of different sensor behaviors, from normal behavior to abnormal behavior, such as a DOS attack (the sensor publishes massive data messages in a short time), node failure (the sensor stops sending data). With data mutation, the TaS enabler can help build datasets for testing many different cases hard to produce in real life. Finally, the *Evaluation* module analyses the

simulation input and output and combines them with the logs collected from the IoT system to provide the final result of a testing process.
The next section presents more details on the functionalities of the tool.

3.2 Tool Implementation

Most of the testing scenarios are defined by the information about the surrounding environment captured by sensors. The following subsection goes into detail about the simulation of sensors.

The simulation

The simulation of sensors — The sensor provides the input data of an IoT system. The simulation of a sensor corresponds to the simulation of the data stream it provides. The simulated sensor has been designed for flexibility in the following ways:

- It supports different types of data report formats:
- It supports different data sources which are used for simulation:
- It supports simulating several abnormal behaviours, such as, low energy, node failure, DOS attack, and slow DOS attack.
- It supports multiple measurements with the different data types, such as Boolean, Integer, Float and Enum. For each measurement, there are several abnormal behaviours that can be selected, such as "fixed value", "value out of range", and "invalid value".

The simulation of actuators — An actuator can be considered as a device that receives the IoT system reaction based on the input data. We simulate the actuator as a component that will receive the reaction signal (actuation data) from the IoT system.

The simulation of a IoT device — In an IoT system, the sensor and actuator are usually part of the same device. An IoT device can contain one to many sensors as well as one to many actuators

The simulation of a network topology — A list of simulated IoT devices forms the simulated network topology. Besides the list of devices, a network topology can also provide the identifier of the dataset (*datasetId*),which contains the data to simulate the SIS in a given time, the global replaying options, the configuration to connect with the database, and the definition of the new dataset where the data generated from the simulations will be stored.

The testing methodologies In this section, we present the testing methodologies and techniques we have adapted in the TaS enabler.

Data Driven Testing — The Data Storage contains the datasets recorded from the IoT system or entered manually. Each dataset contains sensor data (inputs for TaS) and expected actuator outputs. The expected actuator outputs can be the value recorded from the IoT system in a normal scenario. Engineers can also enter them manually via the Graphical Interface. The Evaluation module will use the expected outputs to compare them with the simulation output to determine if they match. A test case passes if the simulation output is within the range of the expected output. The Data-Driven Testing method is suitable for functional and regression testing.
The Data-Driven Testing has been implemented as the main testing methodology of the TaS enabler.

Data Mutation Testing — The Mutant Generator generates new sensor data from existing data stored in the Data Storage by applying one or many mutated functions, such as "change the event order", "change a value", and "delete an event". The mutated data are input for the simulation. The Evaluation module generates a report about the output differences when testing the system with the mutated and the original input data. The Data Mutation Testing method is for penetration, robustness, security, and scalability testing (e.g., mutating the device identifier to obtain new devices). In the TaS enabler, we can mutate the device identity to generate many devices while testing the system scalability. There is also an interface to apply some mutation functions to a dataset manually.
Model-Based Testing [20] and Risk-Based Testing [12] are two other methodologies that we have studied but not yet implemented in the TaS enabler at the time of writing of this paper.

The Testbeds

The Data Recorder — The TaS enabler provides the possibility to simulate an IoT system using historical data.The data is used to create a model that represents the behavior of the physical controlled process. This way, the model works as a digital copy of the physical components. To this end, a Data Recorder module is needed. The Data Recorder records all the events in the real system (coming from the broker). This data (including both sensor and actuator data) is stored in the Data Storage as a dataset. The sensor data can be forwarded directly to the testing system (using the forwarding broker). The more data from sensors are recorded, the more test scenarios are tested. By synchronizing the Sensor simulator timestamps with the Data Recorder, it is possible to simulate a particular SIS (following the DT concept). By monitoring the SIS input and output, we can build an automatic testing process for a complex IoT system.

The Regular and Malicious Data Generator — When testing the IoT system, there are many testing scenarios and cases that do not frequently occur in reality. With the real IoT system, it is almost impossible to collect the datasets for many testing scenarios. The Regular and Malicious Data Generator module helps developers to create a testbed which contains sensor data for various scenarios, e.g., making the temperature too high or too low. By combining multiple data, one can create a testbed that includes many incidents or attack scenarios, such as DDoS and data poisoning. The Data Storage stores all the generated data for further use.

4 Experimentation and Validation

This section presents the application of TaS in a use case which provided by INDRA company[12].

Overview The rail domain requires infrastructure and resources that are usually expensive and require a long-time planning and execution. Therefore, the usage of the rail systems must be highly optimized, following strict security and safety regulations. Several functionalities could be implemented within the rail systems to ensure that the system could tackle its high critical requirements as planned.
The implemented measurements to track and keep a safe behavior of the rail system are developed by the Intelligent Transport Systems (ITS) Domain Use Case. This Use Case will describe logistic and maintenance rail activities. The focus of the demonstrator is placed in the logistics activities.
A Logistics and Maintenance scenario is defined with the aim to provide information of the wagons that conform the rolling stock to assure the well-functioning of the system. These events are only possible through the confirmation of the train integrity, when the different wagons are locked and moving together. This situation ensure the proper transportation of the rolling stock, avoiding possible accidents. A representation for an architecture of this scenario is shown in the Figure 2.

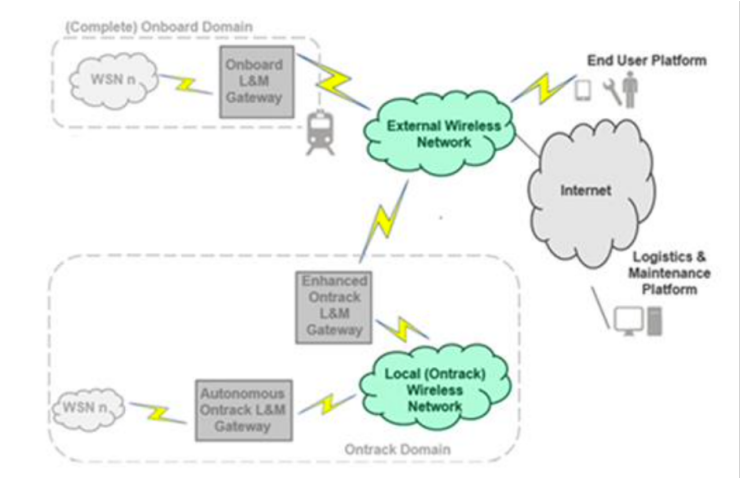

Fig. 2: Wireless Sensor Network Architecture. Source: INDRA

In the figure 2, the ITS system is located in the Logistics and Maintenance Platform, it receives and handles the data provided from the train (OnBoard) and from the track (OnTrack). On each train, there is a WSN (Wireless Sensor Network) which includes several sensors, such as: accelerometer, ultra sound sensor,

[12] https://www.indracompany.com/en/

RSSI (Received Signal Strength Indicator) detector, GNSS (Global Navigation Satellite System) receivers, RFID (Radio Frequency Identification), Humidity, Temperature, CO2 concentrations, Title Detectors; actuators: LED and Display. One OnBoard gateway on each train to send sensor's data to and receive actuated data from the ITS system on the Cloud. The WSN on track contains only a single sensor: RFID. An OnTrack gateway send sensor's data to the ITS system on the Cloud.
The DevOps role in the Use Case consists in providing useful tools to manage the behavior of the different rail components through SW tools. One of them was TaS which focus on simulating and testing the ITS system on two aspects:

- To ensure the ITS handles properly all kind of input data, such as: normal input data, malformed input data, invalid input data, etc.
- To ensure the ITS is able to handle a large number of trains.

4.1 Application of TaS to ITS use case

Figure 3 presents the TaS-ITS integration architecture. EDI (Elektronikas un datorzinātņu institūts, Lavia) provides a testing train on which there are 13 sensors in total. The OnBoard gateway on the testing train connect with the Partners Gateway in INDRA infrastructure. The Partners Gateway receives the input data, then do some validation and pre-processing, the final data is forwarded to the Central Gateway. The TaS tool located in Montimage infrastructure, connects with the Partners Gateway and the Central Gateway to provide three main functions:

- Use a recorder model to record the Partners Gateway data, the recorded data is stored in a Data Storage.
- Use a simulation model to simulate the behaviors of a train based on historical data which was recorded and stored in the Data Storage.
- Use a recorder model to monitor the status of the Central Gateway. The metrics on the Central Gateway are the key values to evaluate the performance of the ITS system.

4.2 Results

The tests are divided in two stages. At the first stage, a recorder model has been used to record the normal behaviors of a single train. The second stage consists on re-injecting the recorded data to perform some tests:

- Scalability testing: adding some scaling factor to check if the gateway can deal with a specific number of trains.
- Penetration testing: adding some data mutation to check if the gateway can deal with some invalid data such as: malformed, invalid value.

The recording stage is shown in the following figures. The figure 4 shows the status of the Central Gateway before activating the train. At this phase, there is no sensors data, however there are still some messages which are the internal messages of the gateways. The figure 5 shows the state of the Central Gateway after activating the train. As shown on the figure, the traffic peak is recorded, the published rate is around 170 messages/second, and the received rate is around 150 messages/second. This traffic is constant as it is required by the the safety system. The recorded data is stored in the Data Storage, then it can be used as an input of a simulation, or it can be duplicated, then mutated to generate a new testing data-set.

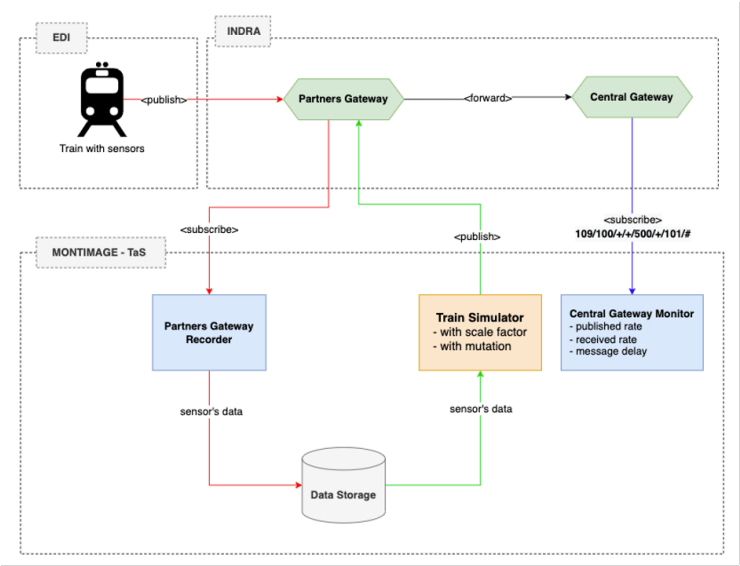

Fig. 3: The TaS-ITS integration architecture

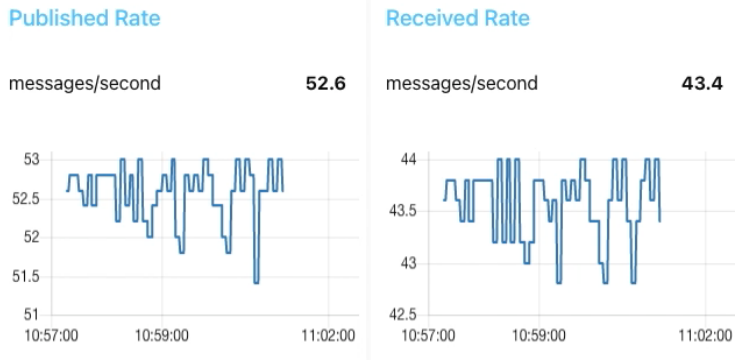

Fig. 4: Before activating the train

Functional Testing — The table 1 has shown the result of functional testing, as it must be noted that the system can handle malformed and invalid value data without crashing.

Mutation Operation	Partners Gateway	Central Gateway
add a valid data row	processed and forwarded	received
delete an existing data row	operators as normal	operators as normal
modify - malformed data	dropped the modified data row	did not receive
modify - invalid data	dropped the modified data row	did not receive

Table 1: Functional Testing result summary

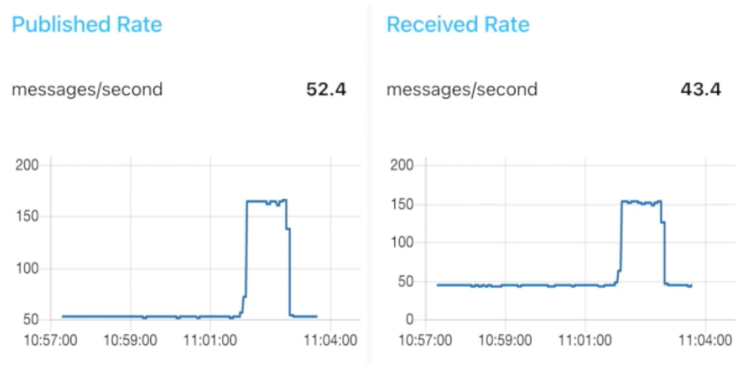

Fig. 5: After activating the train

Scalability Testing — For scalability testing, several scale factors have been tested as shown in table 2

Scale factor	Number of simulated trains	Total number of sensors
1	1	13
5	5	65
10	10	130
20	20	260
50	50	650

Table 2: Scale factors in scalability testing

The figure 6 shows that with the scale factors of 1, 5, 10 and 20, the messages are carried without any issues, the metrics of the Central Gateway are scaled up with a ratio almost the same with the scale factor. However, with the factor of 50, the Central Gateway started to show some delays and the outputs messages are not the same as the input messages, there are some failure indicators which means the ITS Gateway is starting to queue the messages.

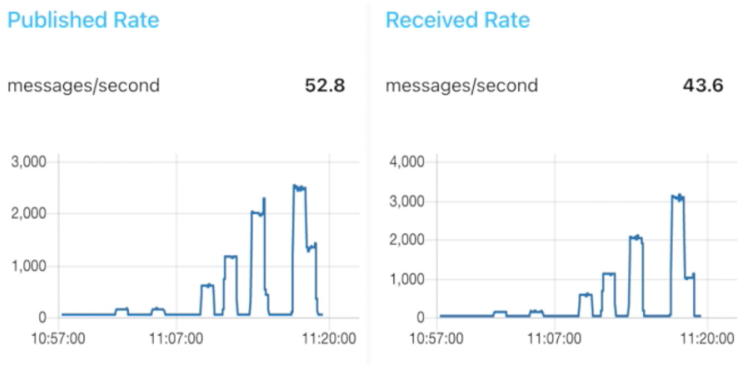

Fig. 6: Scalability Testing result

5 Discussion

The proposed TaS tool helps testing new IoT applications and overcome the simulation limitations. For that it uses data from the real system to create a model of the behavior and create functional, security and stress tests. The tool allows to improve the detection capabilities by automating several steps (e.g., test execution). But still test generation can be improved to cover relevant test scenarios according to defined objectives (e.g., functional testing, regression testing, performance or security testing etc.). The automation of this task will allow to reduce the time of testing the target system as well as improving its coverage.
In the same way, an analysis of real system traces and logs can be used to automate its model building. This reverse engineering task is a complex task that can be also explored as an enhancement of TaS tool.

6 Conclusion and Future Work

In this paper we presented a Digital Twin based tool for an IoT environment, named Digital Twin Test and Simulation tool(TaS). The main objective of this Digital Twin tool is to detect and predict failures in real IoT environments. The TaS tool has been applied in different domains (e-health, transport, telecommunications, smart houses, etc.) showing that the proposed solution is generic and can be applied to achieve different test objectives: security, scalability, energy, etc. In the future, it will be adapted and used in several other collaborative projects dealing with other domains and contexts. To illustrate its application we present a case study, an Intelligent Transport System(ITS) application that provides a simulation of a rail system describing logistic and maintenance activities. Experiments show that our Digital Twin is closely linked to the real world. We can say that both worlds, the real and the digital one are synchronised. In practice, it can help the IoT application developer save time and money on setting up the testing environment and, thus, allows faster delivery of the applications.

ACKNOWLEDGEMENTS

This paper has received funding from the European Union's H2020 Programme under grant agreement no 780351 for the ENACT project as well as grant agreement no 101021668 for the PRECINCT project. Thanks are also addressed to INDRA team that contributed to the experimentation.

References

1. Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., Watteyne, T.: Fit iot-lab: A large scale open experimental iot testbed. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). pp. 459–464 (2015). https://doi.org/10.1109/WF-IoT.2015.7389098
2. D'Angelo, G., Ferretti, S., Ghini, V.: Distributed hybrid simulation of the internet of things and smart territories. CoRR **abs/1710.04252** (2017), http://arxiv.org/abs/1710.04252

3. Faber, F.: Testing in devops. In: Goericke, S. (ed.) The Future of Software Quality Assurance, pp. 27–38. Springer (2020). https://doi.org/10.1007/978-3-030-29509-7_3, https://doi.org/10.1007/978-3-030-29509-7_3
4. Fedorko, G., Molnár, V., Vasil', M., Salai, R.: Proposal of digital twin for testing and measuring of transport belts for pipe conveyors within the concept Industry 4.0. Measurement **174**, 108978 (Apr 2021). https://doi.org/10.1016/j.measurement.2021.108978, https://linkinghub.elsevier.com/retrieve/pii/S0263224121000154
5. Fuller, A., Fan, Z., Day, C., Barlow, C.: Digital twin: Enabling technologies, challenges and open research. IEEE Access **8**, 108952–108971 (2020). https://doi.org/10.1109/ACCESS.2020.2998358
6. Grieves, M.: Origins of the digital twin concept (08 2016). https://doi.org/10.13140/RG.2.2.26367.61609
7. Gupta, H., Dastjerdi, A.V., Ghosh, S.K., Buyya, R.: ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments. CoRR **abs/1606.02007** (2016), http://arxiv.org/abs/1606.02007
8. Kamath, V., Morgan, J., Ali, M.I.: Industrial iot and digital twins for a smart factory : An open source toolkit for application design and benchmarking. In: 2020 Global Internet of Things Summit, GIoTS 2020, Dublin, Ireland, June 3, 2020. pp. 1–6. IEEE (2020). https://doi.org/10.1109/GIOTS49054.2020.9119497, https://doi.org/10.1109/GIOTS49054.2020.9119497
9. Khan, A., Dahl, M., Falkman, P., Fabian, M.: Digital Twin for Legacy Systems: Simulation Model Testing and Validation. In: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE). pp. 421–426 (Aug 2018). https://doi.org/10.1109/COASE.2018.8560338, iSSN: 2161-8089
10. Lee, J., Kang, S., Chun, I.G.: mIoTwins: Design and Evaluation of mIoT Framework for Private Edge Networks. In: 2021 International Conference on Information and Communication Technology Convergence (ICTC). pp. 1882–1884. IEEE, Jeju Island, Korea, Republic of (Oct 2021). https://doi.org/10.1109/ICTC52510.2021.9621144, https://ieeexplore.ieee.org/document/9621144/
11. Leng, J., Wang, D., Shen, W., Li, X., Liu, Q., Chen, X.: Digital twins-based smart manufacturing system design in Industry 4.0: A review. Journal of Manufacturing Systems **60**, 119–137 (Jul 2021). https://doi.org/10.1016/j.jmsy.2021.05.011, https://linkinghub.elsevier.com/retrieve/pii/S0278612521001151
12. Matheu-García, S.N., Hernández-Ramos, J.L., Skarmeta, A.F., Baldini, G.: Risk-based automated assessment and testing for the cybersecurity certification and labelling of iot devices. Computer Standards & Interfaces **62**, 64–83 (2019). https://doi.org/https://doi.org/10.1016/j.csi.2018.08.003, https://www.sciencedirect.com/science/article/pii/S0920548918301375
13. Minerva, R., Lee, G.M., Crespi, N.: Digital twin in the iot context: A survey on technical features, scenarios, and architectural models. Proc. IEEE **108**(10), 1785–1824 (2020). https://doi.org/10.1109/JPROC.2020.2998530, https://doi.org/10.1109/JPROC.2020.2998530

14. Muhissen, M., Shaikh, N., Salah, Z.: Digital twin in artificial intelligence empowerment pisiq. The Open Artificial Intelligence Journal **2** (07 2018)
15. Park, K.T., Nam, Y., Lee, H., Im, S., Noh, S.D., Son, J., Kim, H.: Design and implementation of a digital twin application for a connected micro smart factory. International Journal of Computer Integrated Manufacturing **32**, 1–19 (04 2019). https://doi.org/10.1080/0951192X.2019.1599439
16. Saad, A., Faddel, S., Youssef, T., Mohammed, O.A.: On the implementation of iot-based digital twin for networked microgrids resiliency against cyber attacks. IEEE Trans. Smart Grid **11**(6), 5138–5150 (2020). https://doi.org/10.1109/TSG.2020.3000958, https://doi.org/10.1109/TSG.2020.3000958
17. Sanchez, L., Muñoz, L., Galache, J.A., Sotres, P., Santana, J.R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., Pfisterer, D.: Smartsantander: Iot experimentation over a smart city testbed. Computer Networks **61**, 217 – 238 (2014). https://doi.org/https://doi.org/10.1016/j.bjp.2013.12.020, special issue on Future Internet Testbeds – Part I
18. Sotiriadis, S., Bessis, N., Asimakopoulou, E., Mustafee, N.: Towards simulating the internet of things. In: 2014 28th International Conference on Advanced Information Networking and Applications Workshops. pp. 444–448 (2014). https://doi.org/10.1109/WAINA.2014.74
19. Tao, F., Zhang, H., Liu, A., Nee, A.Y.C.: Digital Twin in Industry: State-of-the-Art. IEEE Transactions on Industrial Informatics **15**(4), 2405–2415 (Apr 2019), https://ieeexplore.ieee.org/document/8477101/
20. Tappler, M., Aichernig, B.K., Bloem, R.: Model-based testing iot communication via active automata learning. In: 2017 IEEE International Conference on Software Testing, Verification and Validation (ICST). pp. 276–287 (2017). https://doi.org/10.1109/ICST.2017.32
21. VanDerHorn, E., Mahadevan, S.: Digital Twin: Generalization, characterization and implementation. Decision Support Systems **145**, 113524 (Jun 2021). https://doi.org/10.1016/j.dss.2021.113524, https://linkinghub.elsevier.com/retrieve/pii/S0167923621000348
22. Zeng, X., Garg, S.K., Strazdins, P., Jayaraman, P.P., Georgakopoulos, D., Ranjan, R.: Iotsim: A simulator for analysing iot applications. Journal of Systems Architecture **72**, 93–107 (2017). https://doi.org/https://doi.org/10.1016/j.sysarc.2016.06.008, https://www.sciencedirect.com/science/article/pii/S1383762116300662, design Automation for Embedded Ubiquitous Computing Systems