Implementation of Digital Twin concept using Microsoft Azure platform

Domagoj Bazina
Ericsson Nikola Tesla
Split, Croatia
domagoj.bazina@ericsson.com

Toni Mastelic
Ericsson Nikola Tesla
Split, Croatia
toni.mastelic@ericsson.com

***Abstract*—Over the last few decades humanity is thriving in terms of digitalization, mostly due to the business or functional requirements. Digital Twin (DT) is one of the concepts that composes virtual world along with the real world, as it features virtual representation of the real life objects that are updated in real time. In addition to the mentioned features, DT generates simulations, learns from them and finally creates decisions based on gathered knowledge. Therefore, it could be exploited in numerous industries and environments. In this paper, a procedure of development and implementation of Digital Twin concept in the Microsoft Azure platform is described.**

***Index Terms*—Virtual, Digital Twin, Microsoft Azure, IoT Hub**

I. Introduction

Term virtual is omnipresent in today's world, especially when we talk about technology and computer science. There is an abundance of concepts and inventions that include aspects of virtual, like virtual reality, virtual machines, virtual assistants, virtual networks and so on. One of the most fascinating features of the "virtual" is its ability to do anything we want, with minimal or non-existing consequences. Moreover, we can create experiments, simulations and learn from them in order to react accordingly as they would occur in the real-world.

Digital Twin (DT) perfectly fits in, as it features an option to design virtual models based on physical entities, like factories, workshops and similar environments [4]. Moreover, DT provides capability to create simulations that would consume time and money in real-world, besides potential hazardous consequences [1]. Input data sourcing from the sensors and actuators is fundamental regarding DT setup as it represents the state of the real-world system at that particular moment.

In order to create a complete DT environment, Microsoft Azure services are used, as they provide variety of simple serverless functions, up to more complex resources such as Digital Twin, IoT Hub and Web Apps [3]. Main goal of this paper is to create a complete setup based on DT resource, that incorporates logic and behaviours that are not originally implemented in Microsoft Azure resources themselves.

Rest of the paper is structured as follows. Section II describes IoT Hub and corresponding resources, Section III gives details about Digital Twin and Section IV includes features about Angular application used as a web interface. Finally, Section V gives a conclusion.

II. IoT Hub Service

IoT Hub service may be observed as an input segment of the whole setup, although it incorporates logic upon which whole setup relies on. Mentioned service (Figure II) consists of 6 resources that include **Microsoft Azure IoT Hub** along with virtual representation of devices, **Azure Time trigger functions** used for device simulations, **Azure Device Provisioning Service (DPS)**, **Azure Event trigger function** that is based on events from the IoT Hub and two **Azure HTTP trigger functions** responsible for managing the whole service.

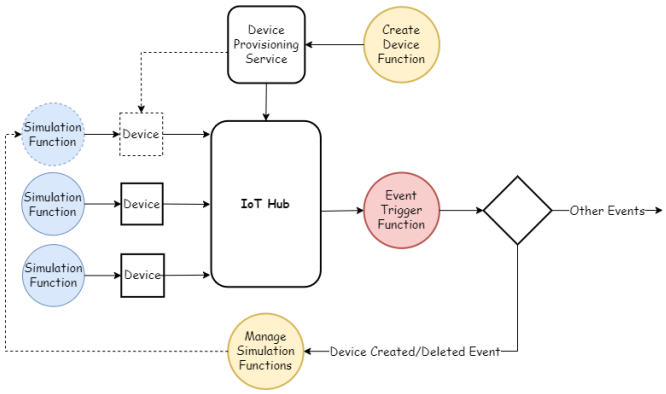

Fig. 1. Structure of the IoT Hub service

IoT Hub is the vital resource as it is an endpoint for devices and it is also a source of different events that are crucial for the rest of the setup. Simulation functions are used instead of real-life sensors. They are configured to execute every 5 seconds. Moreover, each function corresponds to a specific device instance in the IoT Hub. Device Provisioning Service (DPS) behaves like a mediator between IoT Hub and HTTP function intended for registering/deleting devices in the Hub. As previously mentioned, IoT Hub provides several essential events and a role of the Azure Event trigger function is to capture those events and react to them accordingly. There are 3 types of the events in the IoT Hub regarding devices:

1) **Device created/deleted** - whenever new device registers to the IoT Hub, or existing device gets deleted, Event trigger function will make call to the HTTP trigger function for managing device simulation functions.

2) **Device connected/disconnected** - these events are important to notify DT about the state of a certain device.
3) **Device telemetry** - most frequent event that represents data emission from the device.

Event trigger function reacts to these events distinctively, however ultimate purpose of this function is to update the state of the corresponding DT instance based on the information gathered from these events.

III. Digital Twin Service

Digital twin service is the core element of the whole setup, as it is a virtual representation of the real-life system. Furthermore it is a place where the data is processed, simulations executed and decisions generated based on machine learning algorithms [2]. Resources that represent this service (Figure III) include, **Microsoft Azure Digital Twin**, **Azure Event trigger function** based on the events from the DT resource, **Azure HTTP Digital Twin SDK function** intended for managing DT resource and **Azure HTTP Digital Twin Behaviour** function that can be observed as the place where decisions are made regarding DT resource.

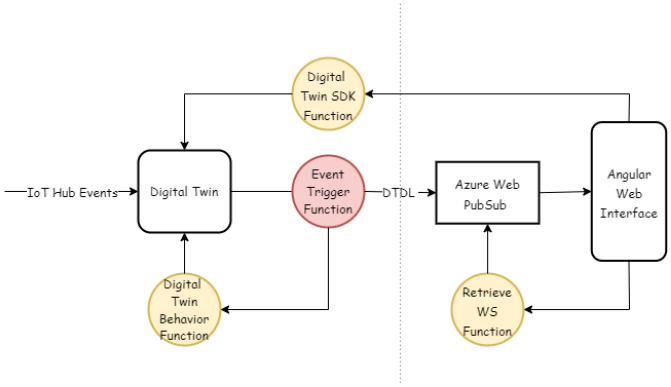

Fig. 2. Structure of the Digital Twin and Web Interface service

Digital Twin resource incorporates DT Models written in Digital Twin Definition Language (DTDL), DT instances and relationships between instances. Furthermore, DT provides multiple events that represent any kind of change in the current setup. Those events are used to improve existing or even to generate new features. Referred events are straightforward and incorporate occurrences such as:

1) **DT instance created/deleted/updated** - event type that occurs whenever DT instance are modified.
2) **Relationship created/deleted/updated** - event related to management of relations between DT instances.
3) **DT telemetry message notifications** - events that represent sensor readings, they are not stored on DT [3].

Relationships play a key role in the Azure DT instances, as the existing ones can not get any new features that are not defined in the DTDL models. Therefore, DTDL models are designed in a way suitable for DT instances to have multiple relationships. SDK function includes set of sub-functions for managing and validating DTDL models, instances and relationships.

Event Trigger function is identical to the one described in the IoT Hub section II, however each event trigger is sent to the Behaviour function designed to enhance features of the DT instance. Enhancing features implies creating simulations, learning from them and designing new features from the existing properties. For instance, if there is a thermometer and humidity sensor in a room, it determines based on particular parameters whether the room is comfortable to stay in or not. Moreover, after each event, function forwards the whole DT tree in DTDL format to the Web interface of the setup.

IV. Web Interface Service

Web interface service is the output segment of this setup, as it provides user-friendly interface intended for managing complete setup. Central resource of the service (Figure III) is Angular application, that is designed independently of Microsoft Azure and deployed using **Microsoft Azure Web App Service**, other resources include **Azure Web PubSub Service** and **Azure HTTP Retrieve URI function**.

Following event triggers from the Digital Twin, DTDL is published through the Azure Web PubSub, service that incorporates real-time messaging using Web Sockets and publish-subscribe model [3]. Angular application uses Retrieve WS function to acquire Web Socket through which data is being published. After establishing successful connection through the Web Socket, Angular app receives DTDL that is thoroughly parsed and stored as the configuration setting. Latter is utilized in order to generate interface that represents state of the DT instances. Along with the representation state, interface provides capabilities for modifying instances through the Digital Twin SDK function requests.

V. Conclusion

Microsoft Azure is a cloud platform that features plethora of services and resources concerning the Digital Twin concept. However, Azure does not provide all resources that are mandatory in favour of implementing DT along with its entire capabilities. Therefore, one of the main efforts of this paper was to enhance existing resources along with their features in order to successfully incorporate the DT into the Microsoft Azure environment. By integrating mentioned resources and services along with the Digital Twin concept, we have created composite system that may be applied in the industry sector, therefore having immense impact on it.

References

[1] Di An and YangQuan Chen. Digital twin enabled methane emission abatement using networked mobile sensing and mobile actuation. In *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pages 354–357, 2021.

[2] IBM. https://www.ibm.com/topics/what-is-a-digital-twin. Accessed: 2022-08-23.

[3] Microsoft. https://docs.microsoft.com/en-us/azure/?product=popular. Accessed: 2022-08-22.

[4] Zhishu Zhang, Jianfeng Lu, Luyao Xia, Sheng Wang, Hao Zhang, and Rongyong Zhao. Digital twin system design for dual-manipulator cooperation unit. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 1431–1434, 2020.