



HAL
open science

Digital Twin – Integrating Cloud Services into Communication Protocols

Thomas Dasbach, Eduardo De Zancul, Klaus Schützer, Reiner Anderl

► **To cite this version:**

Thomas Dasbach, Eduardo De Zancul, Klaus Schützer, Reiner Anderl. Digital Twin – Integrating Cloud Services into Communication Protocols. 16th IFIP International Conference on Product Life-cycle Management (PLM), Jul 2019, Moscow, Russia. pp.283-292, 10.1007/978-3-030-42250-9_27 . hal-04444761

HAL Id: hal-04444761

<https://inria.hal.science/hal-04444761v1>

Submitted on 7 Feb 2024

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

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



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Digital Twin – Integrating cloud services into communication protocols

Thomas Dasbach¹, Eduardo de Senzi Zancul², Klaus Schützer³ and Reiner Anderl¹

¹ Technische Universität Darmstadt, 64283 Darmstadt, Germany
dasbach@dik.tu-darmstadt.de
anderl@dik.tu-darmstadt.de

² University of São Paulo, São Paulo, Brazil
ezancul@usp.br

³ Methodist University of Piracicaba, São Paulo, Brazil
schuetzer@scpm.unimep.br

Abstract. The Digital Twin is the following development of ongoing improvements in communicational technologies, decreasing hardware costs and the need for products with decentralized decision-making abilities. In order to improve the qualities of decisions, there is the requirement for an easy, reliable way to transport data between Digital Twins, even if the network connection has a low bandwidth or is unstable. As a solution, the use of a cloud service as a broker architecture is described. A concept for communication architecture is given as well as the required design of a Digital Twin. The proposed concept has been implemented to assess the resulting advantages and limitations.

Keywords: Digital Twin, Cloud storage, Communication protocol, Product Lifecycle Management

1 Introduction

Industry and society are always subject to change. One of the major developments of nearly the last two decades has been the ongoing digitalization. Several factors, like the increased availability of wireless communication, enhanced user interfaces and arise of new technologies like cloud services led to a point where the demand of products and machines, which are to a small degree self-aware and able to decide for themselves for the most suitable usage, rises [1].

To enable reliable decisions, it is crucial to know everything or at least as much as possible related to the matter [2]. In the case of products, an approach to aggregate most of the data can be seen in Digital Twins. Over time, integrating modern information, communication, and sensor technologies developed into complex constructs, which are enhanced by simulation models and databases. They are a sufficient starting point to develop the goods of the future [3].

In order to further advance the resilience of the decisions, the amount of available data for each Digital Twin has to be increased beyond the point of just using only the sensors connected to the specific product to a scenario where each Digital Twin can benefit from all sensors within the near perimeter of the product. Therefore it is necessary to exchange large sets of data between them. This can happen either with peer-to-peer connections or, as in the concept described, with the usage of cloud services as a broker. In addition to the advantage of a temporally decoupled data transfer, the actual workload for each Digital Twin is reduced: if more than one request is made, the requirement for the hardware of each Twin is lowered.

The objective of this paper is to present a concept proposal for enhancing data transfer between Digital Twins based on cloud services. The proposed concept is detailed and implemented in a test scenario to support assessment.

The remainder of this paper is structured as follows: Section 2 presents a literature review regarding the Digital Twins and communication protocols, followed by Section 3, which submits the proposed concept. Section 4 describes the pilot implementation. Finally, Section 5 presents the main conclusions and suggestions for further research.

2 Literature

In the following passages, the underlying key requirements of the concept will be discussed. Especially chapter 2.2 should be regarded as only a summarized segment of a large field of study with a focus on technologies which are used in the development of the proposed concept.

2.1 Digital Twin

Historically, the concept of an exact object's copy to study its real counterpart behavior can be traced back to the NASA's Apollo program, where two identical space vehicles were made. One was sent to the mission, and the other one was kept on Earth to allow mirroring the conditions faced by the mission's vehicle as accurately as possible. This monitoring was crucial to achieving a better understanding of the critical situations faced and help the astronauts more efficiently [4, 5].

This first Twin concept for monitoring a system or object, even if purely physically, was of great importance to inspire the development of software-based Digital Twins. The software-based Digital Twin has been used to monitor the real-time condition of a connected Physical Twin through modern sensing, communication, and identification technologies, such as RFID, QR-Code, Machine Vision, etc. and to validate the gathered data with the virtual model of the physical counterpart [6–9].

Over time, especially with the upcoming of *Industrie 4.0*, the focus of the concept Digital Twin shifted towards production and manufacturing of smart products [10]. The term Digital Twin has been defined in many attempts by several authors but each one with a slightly different focus. Most of them describe the Digital Twin as a comprehensive collection of information gathered throughout the entire product lifecycle [3]. A

different trend is followed by Boschert and Rosen, who define the Digital Twin as only an aggregation of the currently relevant data [11].

In this paper, the Digital Twin is understood as an accumulation of all gathered data, simulation models enhanced by the real environmental influences and own decision trees to adapt the behavior of the physical counterpart regarding the perceptions from simulations and similar products in alignment to previously given definitions like the one of Haag and Anderl [3].

2.2 Communication protocols

Concerning the communication protocol, there is a variety of available solutions at the market; most of them with advantages and disadvantages depending on the particular use case. Therefore, only a small selection is presented here. MQTT in this case can be taken as an example for several relatively new, lightweight protocols like CoAP, AMQP or STOMP.

In April 1971, the first FTP standard RFC 114 was published. After multiple subsequent RFCs, RFC 959, File Transfer Protocol (FTP) was published in October 1985. Unlike most protocols, the FTP standard does not use one TCP connection, but two. One of them is the Control/Command Connection, the other one is the Data Connection. The former connection is maintained throughout the entire FTP session and used to pass control information, such as FTP commands and replies. The latter one is used for data transfer and is opened and closed as needed [12].

The FTP standard specifies an active and a passive mode. The main difference between the two is whether the client or the server initiates the data connection [13]. Due to its design it inherits some security problems, which are hardly possible to solve, as well as complex handling to set up properly. So newer protocols like HTTP or HTTPS should be taken into account [14].

Released in 1999 by IBM, MQTT (currently Message Queuing Telemetry Transport) exchanges data using packets (or “message formats”) in an organized specific manner [15]. Each of the packets contains three parts: a mandatory fixed size header, an optional header with variable length and a possible payload, again with a variable length [16].

MQTT has two kinds of participants, clients and brokers, and uses a publish-subscribe concept. Clients can be publishers or subscribers, meaning they can publish messages for the interested users, subscribe or unsubscribe in interesting subjects in order to receive messages, attach and detach from the brokers. The brokers control the distribution of information, processing requests from clients, accepting client requests, receiving published messages by clients and sending publisher messages to subscribed clients [16].

HTTP is one of the most widespread protocols within the web. Developed by Tim Berners-Lee, it is a standard since 1997. In contrast to MQTT, it uses an architecture based on clients and servers. Meaning a server will never initiate a connection and a client will never expect an incoming transmission. The server can gather data, given to them by clients, and distribute this data, once a client requests this data.

Originally designed for the web, it has some disadvantages regarding the use in IoT applications. For example, the overhead of HTTP messages is big in comparison to other protocols. HTTP has quite high hardware requirements, high latency, and a big bandwidth compared to other protocols. Despite these facts, HTTP has two advantages which are very interesting: High security standard as well as very high interoperability [16], both basic interests for IoT.

3 Concept development

In order to create a communication protocol for a Digital Twin, the first step is to define the design of the Digital Twin itself. No matter how complex or, more likely, how slim it needs to be, a Digital Twin will always consist of at least three functional parts – frontend, core, and backend. The following described architecture allows easy extensible design of the separate parts. Therefore, it is kept abstract with references to the recommended features. The basic layout with the extension of a cloud unit is visualized in figure 1.

From the perspective of an end customer, the frontend of a Digital Twin will consist of an interface unit, dedicated to general communication to other Digital Twins or customers. A various amount of communication protocols can be used to initiate data transfer. Due to the later-described method to discover other Digital Twins, the previous mentioned MQTT with its client-broker architecture is a suitable solution for the concept described in this paper.

Behind the frontend is the core of the Digital Twin. The core serves as the decision unit; it runs the decision trees, processes the sensor values of the physical counterpart and initiates communication requests to other Digital Twins or web services. Therefore, it uses the functionalities that the interface unit provides. Another key aspect of the core is to request data from other sources and pass this incoming data down to the storage unit. Besides the core, there is the possibility to include further simulation models or additional software, which will be executed by the core, if necessary. The complexity of the local simulations is limited by the hardware of the core. In practice this will result in the approach to reduce local simulations as much as possible in order to decrease costs of products. The lower boundary for the hardware decrease will be the crucial simulations for the uninterrupted running. Which simulations have to be declared as crucial depends again on the network connectivity. If the connection is very stable most simulations can be run elsewhere, otherwise more simulations need to run locally resulting in higher requirements for the hardware.

The backend is a storage unit for system variables. Furthermore, it caches all data provided by sensors and sub-ordinary Digital Twins. Data can be passed to both, the cloud, as well as to the communication interface to send it to other Digital Twins and vice versa. The storage unit can store data locally. The required size of the storage depends on the network stability of the Digital Twin. Whenever there is a disconnect data needs to be stored until it can be uploaded again. Technically the most likely implementation of such a storage unit is created by a local database or if the spare hardware resources are available by a database attached to a server.

To unburden the core unit, the storage should be able to handle the data of the sensors independently. If the values need processing, the core can request them once it is idle. In addition, the incoming data of subordinate Digital Twins can be handled at the storage unit. In case of sufficient storage capacity, there is a variety of available databases to order and process the data for later usage.

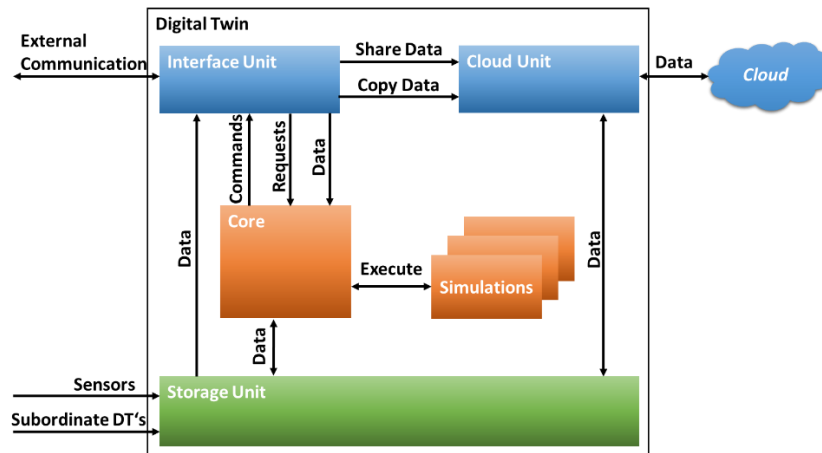


Fig. 1. Digital Twin architecture

Independently of the underlying hardware, the Storage Unit may run out of free space at a point in time. Reasons for that can be an unexpected long usage of the product, peaks in the needed storage space after requesting large data sets from other sources, or extension of sensors later on.

The possible counter-strategies consist mainly of two categories: on the one hand, it is possible to decrease the stored data, and on the other hand to increase the available space. In order to decrease the stored data, it is either needed to decide which data will not be needed in future use cases and to delete those or to compress the stored data. Deleting data is in case of a Digital Twin not a long-sighted strategy because the decision-making criterion cannot take future technologies and their requirements into account. Therefore, as an alternative, the storage space needs to be increased. Increasing the storage space before the deployment of the Digital Twin leads to the difficult decision of the required margin. In order to absorb every possibility, such as the extension of the product by adding additional sensors, it is necessary to aim for a very large margin, which in the end leads to an increase of production costs and needed materials.

It will be difficult to increase the storage space physically at the product, once it is needed, if the product is embedded into larger and inaccessible systems. Such a design would critically limit the usage of the Digital Twin. Thus, it needs to be accessible with a reasonable effort.

Another solution to solve the storage issue is the usage of established cloud services. Therefore, the interface unit is extended, as seen in figure 1, by a cloud unit. The cloud unit handles communication with a contractor's or the manufacturer's cloud. With this

setup, it is possible to add further storage space without physical access to the Digital Twin.

Additional to the solved space problem, this setup enables a second way of transferring data between different Digital Twins. If the large data collection of a Digital Twin gets stored in the cloud, it is possible to transmit data within the cloud.

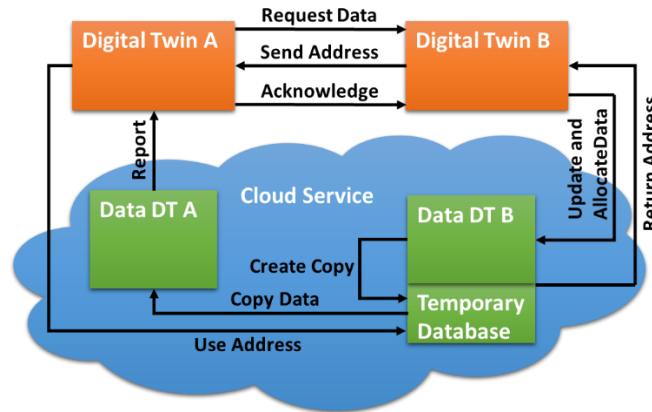


Fig. 2. Digital Twins data transfer based on cloud services

Figure 2 shows the steps included in the proposed data transfer. In this case, the Digital Twins store their data up to a certain degree on a local storage, but whenever possible, they update a data set located in a cloud, which includes all the data gathered over the entire life of the product.

This double storage of information allows another way of transferring the data between two Digital Twins. In the described use case it is presumed that both Twins already know about each other. If Digital Twin A needs data from Digital Twin B, it will send, independently of the later used transportation method, a request to Digital Twin B. This request must already contain information about all available ways of data transfer. So, B can decide depending on varying factors, e.g. size of the requested files or stability of the connection, to use the cloud transfer. In the first steps, if necessary, B updates its data set within the cloud, then the cloud provider creates a temporary copy with the requested data and returns the address and access keys for this set to the Digital Twin. Digital Twin B passes the address and the key to Digital Twin A. With the address and the key, A can order the cloud service to copy the temporary data set into the online storage of A. Once this is finished, a report is sent to Digital Twin A and it passes an acknowledgment to Digital Twin B. Any needed information included in the transferred data can now be exchanged between the cloud service and A.

Another benefit provided by the usage of a cloud infrastructure is the possibility to use an online client as a manager for the addresses of all active Digital Twins. If all Digital Twins try to establish connections to a cloud in order to update their information, it is an easy task to keep track which Twins are active and what kind of services they could provide. Within this concept, all addresses of the participating

Digital Twins were stored in a so-called phone list, in reference to a dissipating phone book.

Basically, the phone list functions as a central database for all available participants of a certain network. Digital Twins that do not store data regularly online have to regularly confirm their activeness towards the phone list, otherwise their information and addresses will be deleted to avoid degeneration of the database.

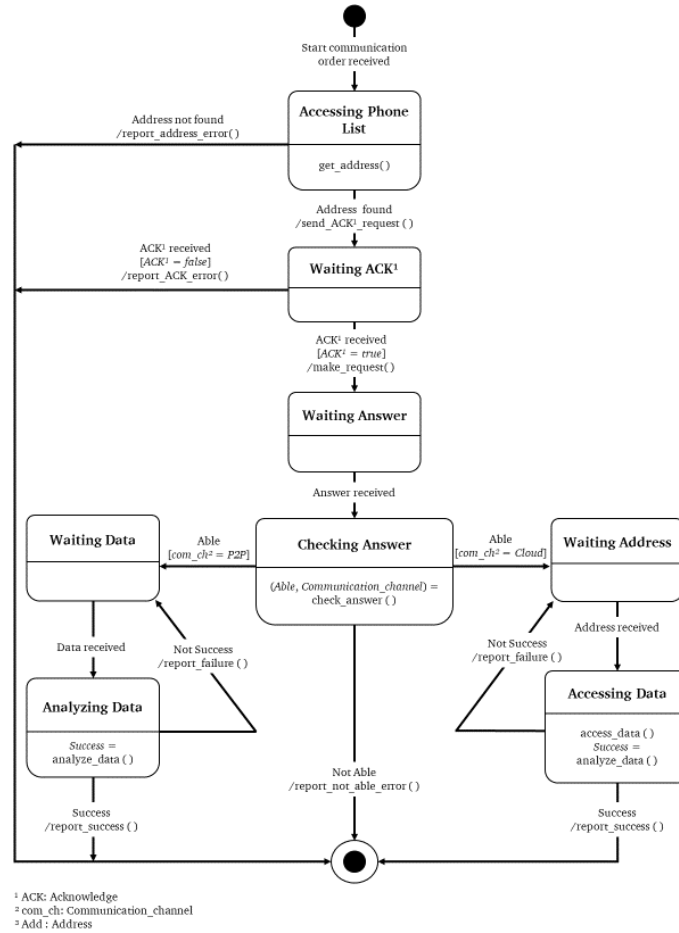


Fig. 3. Flow chart of communication selection [17]

With these parts of the concept, a behavior flow chart for Digital Twins, which shows how a regular data transfer will be ordered was designed, like seen in figure 3. In a first step, the Digital Twin will connect to the phone list to get an address of a specific Twin or of a Twin, that can provide the services needed. After receiving the message it establishes a message via HTTP, due to its high interoperability. First step in communication between both Digital Twins has to be defining which way they will use to transmit the data. For this step, all methods available for both parties have to be

considered. After a final decision, the communication switches to the selected protocol, the data is transferred and the communication channel is closed again.

4 Implementation

In order to test and assess the proposed concept, a pilot scenario has been implemented. For the implementation, it was crucial to use mainly low budget and low tier hardware for the Digital Twins to prove the functionality of the concept, even with limited capabilities.

The prototype was based on sensors and Arduino and RaspBerry Pi microcontrollers in order to test the algorithm with real-time data. The setup is presented in figure 4.

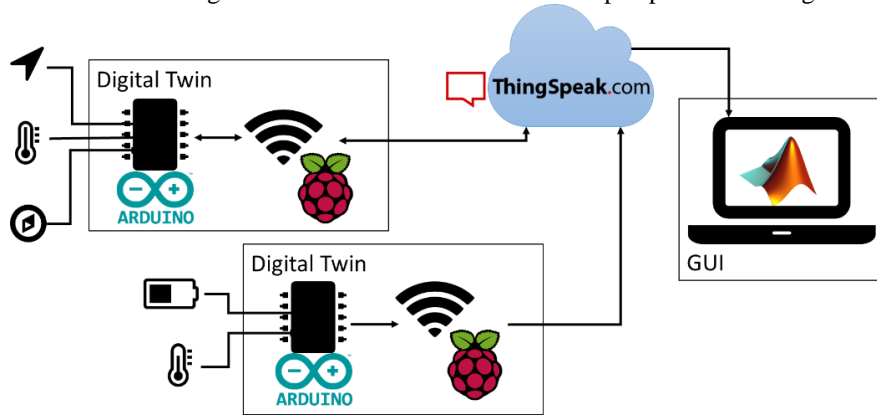


Fig. 4. Implementation for the test scenario [17]

Two Digital Twins were set up, each one made from an Arduino Leonardo and a RaspBerry Pi, which together represent a Digital Twin with its physical and digital counterparts. Each Digital Twin uses several sensors to generate data. The sole purpose of the RaspBerry Pis in this setup is to function as a relay for the messages of the Arduino into the network and therefore don't play a significant role. The Digital Twins connect to the Cloud server *ThingSpeakTM*.

ThingSpeakTM is an IoT cloud platform that allows data aggregation, visualization feature and live data stream analysis. Accessible through MQTT or a REST API it has the ability to execute *MATLAB[®]* code online to perform analysis [18].

The protocol MQTT is used as an exchange protocol between Twins and ThingSpeak, since it is very lightweight compared to HTML and very established compared to other protocols [19]. Messages transferred from the Digital Twins to ThingSpeak are relatively small, because data is transferred regularly and each Twin owns a maximum of three sensors. In case of a more unstable connection, resulting in larger information pieces, it would be useful to switch to a fully HTML transfer. Due to the earlier mentioned security problems, FTP was no suitable solution.

The Digital Twin was able to connect to other Digital Twins in order to receive or send data following the communication protocol. The P2P communication was handled

with TCP/IP in order to avoid a large overhead. If a more extensive data set between the Digital Twins needed to be exchanged, an algorithm decided whether the data transferred directly over the TCP/IP connection or if it would be more suitable to create a new access key on *ThingSpeakTM* and only send it. After the key transmission, the second Twin was able to access the stored data in the cloud and could download all shared information. Depending on what kind of permission the Twin was given, it was able to gather the data later on, if it lost connection or was busy with other tasks.

5 Conclusion

This paper presents a concept for improving Digital Twins communication. It is not the goal of the proposed concept to eliminate P2P communication entirely, but rather to propose a second, more effective way to communicate between Twins, if certain limitations are given. The benefits of the proposed communications are most relevant, if both Digital Twins have an unstable connection while the disconnection between them is not temporally connected. The described concept offers four main advantages, identified during pilot assessment. First, on a design level, the separation between the interface unit and cloud unit allows for an easy switch of the cloud service, because the cloud unit is not an essential part of the Digital Twin and can be exchanged at any time. Second, the transfer of data through the cloud decreases the time each Digital Twin needs for the data transfer, because not both of the Digital Twins need to maintain a stable internet connection at the same time. This does not imply that the data transfer in general takes shorter.

Third, given the fact the Digital Twin, which receives the request, maintained its cloud storage, the actual workload for this Digital Twin is lowered to a minimum. The whole task of reading the data and sending is done by the cloud service, with likely more sufficient hardware. Also, the storage in the cloud enables non crucial simulations being executed in the cloud. And the last benefit results from the use of the central directory for all Digital Twins, due to the easy monitoring and the slim approach for finding other Digital Twins with particular services.

The proposed concept is part of a broader research effort in order to establish the conceptual foundation, concepts, and methods to enable Digital Twin adoption in larger scale in industry. Further suggestions for research in the area include a pilot implementation using real data and extension to multiple Digital Twins communicating production environments, as well as using the accumulated data to run simulations at the cloud to unburden the Digital Twins.

Acknowledgment.

The authors thank Ms. Peixoto Barbosa, Mr. Machado Lima, Mr. Carvalho Gaspar de Barros Bello, Mr. Siguematu, and Mr. Oliveira Françani for detailing the proposals technical aspects and implementing the pilot test scenario as part of a student project.

References

1. Rayes A, Salam S (2017) Internet of Things From Hype to Reality. Springer International Publishing, Cham
2. Iselin ER (1988) The effects of information load and information diversity on decision quality in a structured decision task. *Accounting, Organizations and Society* 13(2): 147–164. doi: 10.1016/0361-3682(88)90041-4
3. Haag S, Anderl R (2018) Digital twin – Proof of concept. *Manufacturing Letters*
4. Rosen R, Wichert G von, Lo G et al. (2015) About The Importance of Autonomy and Digital Twins for the Future of Manufacturing. *IFAC-PapersOnLine*
5. Schleich B, Anwer N, Mathieu L et al. (2017) Shaping the digital twin for design and production engineering. *CIRP Annals - Manufacturing Technology*
6. Uhlemann TH-J, Lehmann C, R. Steinhilper R (2017) The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0. *The 24th CIRP Conference on Life Cycle Engineering*
7. Tao F, Zhang M (2017) Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. *IEEE Access*
8. Brenner B, Hummel V (2017) Digital Twin as Enabler for an Innovative Digital Shopfloor Management System in the ESB Logistics Learning Factory at Reutlingen - University. *Procedia Manufacturing* 9: 198–205. doi: 10.1016/j.promfg.2017.04.039
9. Uhlemann TH-J, Schock C, Lehmann C et al. (2017) The Digital Twin: Demonstrating the Potential of Real Time Data Acquisition in Production Systems. *Procedia Manufacturing* 9: 113–120. doi: 10.1016/j.promfg.2017.04.043
10. Negri E, Fumagalli L, Macchi M (2017) A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manufacturing*
11. Boschert S, Rosen R (2016) Digital Twin—The Simulation Aspect. In: Hehenberger P, Bradley D (eds) *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers*. Springer International Publishing, Cham, s.l., pp 59–74
12. Charles M. Kozierok (2005) *The TCP/IP Guide Version 3.0*
13. Jay Ribak (2006) Active FTP vs Passive FTP: a Definitive Explanation
14. Frank Bernard (2002) Sicherheitsprobleme beim FTP-Protokoll. *Linux-Magazin*
15. A. Banks, R. Gupta (2014) MQTT Version 3.1.1. OASIS Open
16. D. Soni, A. Makwana (2017) A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT)
17. Peixoto Barbosa, Machado Lima, Carvalho Gaspar de Barros Bello et al. (2019) Design of a communication protocol for the use between digital twins. ADP, TU Darmstadt
18. The MathWorks I (2019) Thingspeak. https://thingspeak.com/pages/learn_more
19. Naik N (2017) Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In: *2017 IEEE International Symposium on Systems Engineering: ISSE 2017 : Vienna, Austria, October 11-13, 2017 : 2017 symposium proceedings*. IEEE, Piscataway, NJ, pp 1–7