01 Jun 2018

Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MT Connect

Liwen Hu

Ngoc-Tu Nguyen

Wenjin Tao

Ming-Chuan Leu
Missouri University of Science and Technology, mleu@mst.edu

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/mec_aereng_facwork/4177

Recommended Citation

L. Hu et al., "Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MT Connect," *Proceedings of the 46th SME North American Manufacturing Research Conference (2018, College Station, TX)*, vol. 26, pp. 1193-1203, Elsevier, Jun 2018.
The definitive version is available at https://doi.org/10.1016/j.promfg.2018.07.155

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 26 (2018) 1193–1203

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA

Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MTConnect

Liwen Hu[a], Ngoc-Tu Nguyen[b], Wenjin Tao[a,*], Ming C. Leu[a], Xiaoqing Frank Liu[c], Md Rakib Shahriar[c], S M Nahian Al Sunny[c]

[a] *Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65401, USA*
[b] *Department of Computer Science and Engineering, University of Minnesota Twin Cities, Minneapolis, MN 55455, USA*
[c] *Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701, USA*

Abstract

The common modeling of digital twins uses an information model to describe the physical machines. The integration of digital twins into productive cyber-physical cloud manufacturing (CPCM) systems imposes strong demands such as reducing overhead and saving resources. In this paper, we develop and investigate a new method for building cloud-based digital twins (CBDT), which can be adapted to the CPCM platform. Our method helps reduce computing resources in the information processing center for efficient interactions between human users and physical machines. We introduce a knowledge resource center (KRC) built on a cloud server for information intensive applications. An information model for one type of 3D printers is designed and integrated into the core of the KRC as a shared resource. Several experiments are conducted and the results show that the CBDT has an excellent performance compared to existing methods.

Peer-review under responsibility of the scientific committee of the 46th SME North American Manufacturing Research Conference.

Keywords: Smart Manufacturing; Digital Twin; MTConnect; Cloud Manufacturing.

1. Introduction

Boosted by the evolution of cloud computing and Internet-of-Things (IoT) technologies, manufacturing can be done over any physical border to scale up via the Internet. To overcome the shortcomings of the traditional approach that cannot provide a comprehensive view on the system's condition during the working progress of manufacturing machines, web-based technology has been used to visualize physical machines in the virtual world over the Internet [1–3].

Recently, digital twins are often the best solutions that support remote interaction of humans with physical machines and overcome the challenge of geographical distance [4–8]. To improve the efficiency of remote monitoring and planning of physical machines, there is much ongoing effort to adapt digital twins to a Cyber-Physical Cloud Manufacturing (CPCM) system, which uses a cloud server for central coordination [9]. Since a large number of applications are required to be processed simultaneously, the number of requests through the cloud server may be beyond the available computing capacity [10], including memory and network bandwidth, causing a high burden to the system.

MTConnect protocol [11] has been adapted to manufacturing factories for data acquisition, instead of directly communicating the CPCM system with multiple sensors [12–16]. It reduces the overhead of data acquisition and minimizes the delay time in communications.

---

[*] Corresponding author. Tel.: +1-573-466-3528 ; fax: +1-573-341-6512.
E-mail address: w.tao@mst.edu

2351-9789 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 46th SME North American Manufacturing Research Conference.
10.1016/j.promfg.2018.07.155

The aim of this study is to build cloud-based digital twins (CBDT) with usage predictions and estimation utilities to improve the demands of using resources efficiently. The CBDT aggregates the manufacturing resources through a virtualization interface that has the capability to store and tag each data point to make the data stream traceable. In this paper, a CBDT method is designed with an information model to satisfy the requirements of building a digital twin on a CPCM system. Our method is executed using MTConnect protocol's operating process to reuse its shared resources based on the integration of protocols. The alternate resource utilization secures reliable characterization of data in the system, thereby reducing both load and power consumption. We evaluate the proposed method by implementing it on a practical CPCM system testbed. The results show that the shared data resources reduce the total power consumption and workload of the server as well as guarantee the quality of service.

The remaining sections of this paper are organized as follows. The CPCM concept and MTConnect protocol are introduced in Section 2. The proposed method for constructing digital twins is detailed in Section 3. The simulation results are analyzed in Section 4. Finally, Section 5 summarizes this paper.

2. Cyber-Physical Cloud Manufacturing System Concept and MTConnect

2.1. *Cyber-Physical Cloud Manufacturing system concept*

Monitoring and diagnosing the manufacturing machines through the Internet is a challenging task. The scope of CPCM's application is recently widened by the developments of Industry 4.0 in Europe [17]. A CPCM system is designed to provide manufacturing services as well as allow the machining tools to be monitored and operated directly from the cloud [18,19].

Figure 1 shows how remote manufacturing is conducted within the CPCM system. A web-browser gateway for users can access cloud applications to monitor a machine and plan its operation remotely during the manufacturing process. The requests of access by the users are immediately connected to the private clouds, which may be scatted in specific regions, and then coordinated. Furthermore, a main station that works as a public cloud connects all the private clouds together. The MTConnect protocol is embedded to supply the KRC with the data from physical machines.

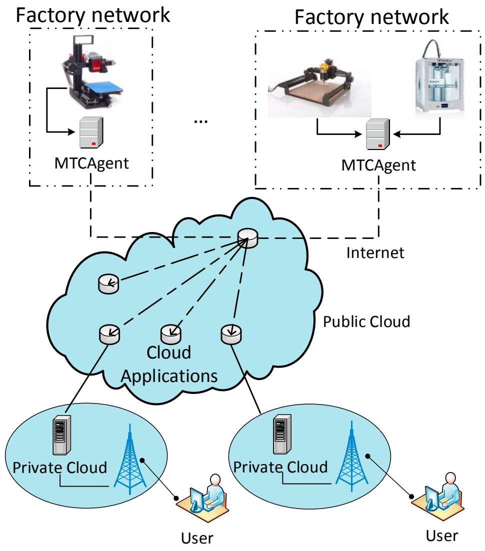


Fig. 1: The concept of Cyber-Physical Cloud Manufacturing system using MTConnect.

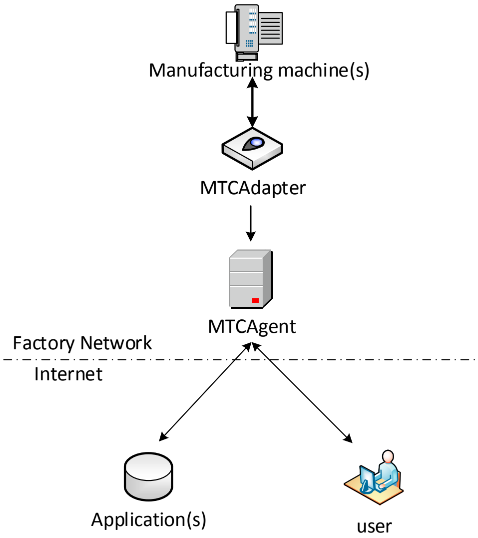


Fig. 2: The working principle of MTConnect protocol.

2.2. *MTConnect*

The MTConnect protocol has become more and more popular since its first version was released in 2008 [11]. The name "MTConnect" stands for "Manufacturing Technology Connect" and the communication protocol was designed for strengthening the data transmission between the various manufacturing resources (including machines on shop floors) and applications. Figure 2 shows the working principle of the MTConnect, where the MTCAgent aggregates the information of machine(s) from MTCAdapter in the factory network. The MTCAgent provides manufacturing information in standard XML format to users over the Internet. Another feature of the MTConnect is to provide manufacturing services with REpresentational State Transfer (REST) or RESTful interface, which allows the client obtaining data without a logon/logoff sequence or establishing any sessions. The MTConnect offers an interface between machines and a manufacturing cloud, which bridges the local manufacturing network and the Internet and assists the cyber-physical manufacturing cloud to reach local manufacturing resources. Moreover, MTConnect uses the XML [20] protocol to describe the data structures and the status of machines. This reduces the usage of cloud computing resources.

3. Cloud Based Digital Twins of Manufacturing Machines

3.1. The motivation

The CPCM system is a hybrid system in which multiple manufacturing factories are connected to a cloud server via the Internet and thousands of manufacturing machines are involved. Every interaction with the physical machines is conducted from the cloud. The services in the cloud system are integrated in the inner cores such as Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) that support all activities of the applications. The relentless requests placed on the cloud server may become a stretched condition. A minor flaw in the connections between factories and the manufacturing cloud can paralyze the related cloud application, resulting in a cascading failure of the CPCM system. For example, once the HTTP service in the cloud server fails because of overload, it will result in an out-of-service of the client-server connection between the manufacturing system and the cloud system. Because of this vulnerability, it is required to minimize the use of bandwidths and computing resources. Therefore, it cannot be conducted by the traditional cyber-physical manufacturing system based digital twin building method.

To address these challenges, we propose a new method to construct a new framework to build digital twins in CPCM systems, called Cloud-Based Digital Twins (CBDT). We will demonstrate the implementation of our method through experimentations on a practical CPCM testbed. The main contributions of this framework are summerized as follows:

- We introduce a knowledge resource center (KRC) in order to redefine the machine's structure as well as coordinate all data streams flexibly during the digital twin building process.
- Our method is executed using the MTConnect protocol to reuse the shared resource based on the integration of protocols.
- Our method reduces the total overhead by using shared resources from KRC to achieve a smooth performance of the digital cloud.

3.2. The KRC framework

The KRC is designed with six modules, including the initializer, parser, information model, data station, digital twin platform, and configuration center. All interactions between the modules in the KRC are connected and configured by the configuration center. The initializer works as a gateway of KRC to support all communications between the physical machines and the KRC. While the data in MTConnect protocol standard is organized by XML's structure, the parser retrieves the data from MTConnect protocol standard, and then the extracted data are forwarded to the data station, where the data are stored in the server for building the digital twins. The configuration center also bridges the information model and the data station and provides memory cells to store processed data. In the last stage, the digital twin platform is set to present the completed digital twins to the users. The traditional method uses manual connections to supply data to construct the digital twin. These manual connections consume a lot of resources, which may result in the failure of system when digital twins are built for different types of manufacturing machines at the same time. In contrast, the CBDT uses the information model as a central resource in the KRC and serves as a repository and shareable resource for every input of any requests for building digital twins. These features plus functional modules allow the connection of physical machines to the KRC without any manual influences.

Shown in Figure 3, any requests of users sent to the cloud sever are immediately coordinated by the private cloud and forwarded to the configuration center in the KRC. Once receiving the requests, the configuration center activates the initializer as the beginning of the digital twin building process. The initializer works as an anchor to connect the KRC to a physical machine in the

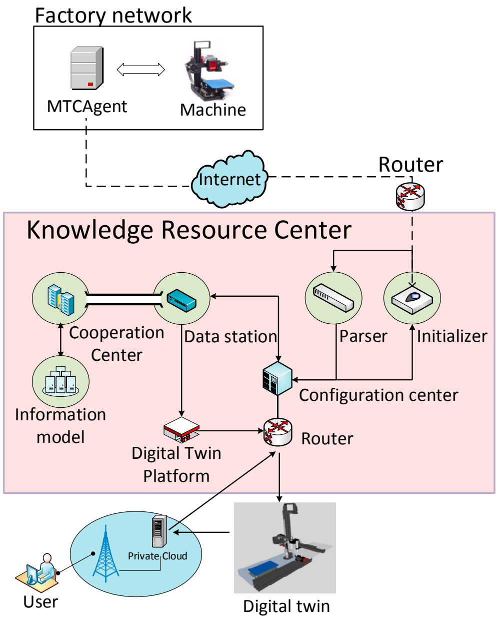
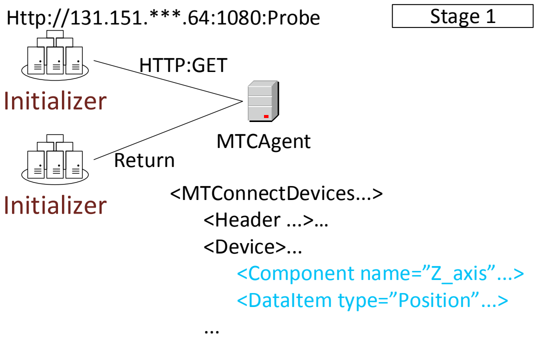


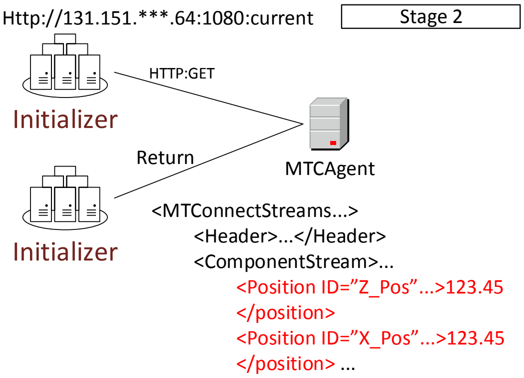
Fig. 3: System architecture of cloud based digital twins of manufacturing machines

factory network through the Internet. This connection process is divided into two stages described below. In the first stage, because the KRC needs to create a guide model, which requires virtual components, the initializer sends requests to get descriptive data from physical machines to let KRC determine what virtual components should be included in the guide model. All responses from the physical machines to the KRC through the initializer are aggregated to an XML file. These requests in the first stage are defined by a specific type of command known as the *Probe* command. This command provides a mechanism for extracting the information of a machine's structure from the physical machine. This information is then used to construct the data for the responded XML file.



```
<MTConnectDevices...>
    <Header ...>...
    <Device>...
        <Component name="Z_axis"...>
        <DataItem type="Position"...>
    ...
```

(a) Stage 1: The probe command and MTCAgent response.



```
<MTConnectStreams...>
    <Header>...</Header>
    <ComponentStream>...
        <Position ID="Z_Pos"...>123.45
        </position>
        <Position ID="X_Pos"...>123.45
        </position> ...
```

(b) Stage 2: The current command and MTCAgent response.

Fig. 4: The initializer's inquiry commands and their corresponding responses.

3.2.1. *Initializer and MTConnect communication*

In the second stage, the KRC requests updated information of the physical machine's components, such as the position, rotation, and speed to the corresponded components in the digital twins. In this stage, the initializer sends the *Current* command to the factory MTCAgent and inquires the current status of each component in the physical machine. All communications between the initializer and the MTCAgent are conducted and transported by the Hypertext Transfer Protocol (HTTP). The details of the two types of commands [11] used by the initializer to communicate with the MTCAgent are as follows:

- *Probe command* to retrieve the component's information and the data items for the system. It returns an MTConnectDevices XML document.
- *Current command* to retrieve a snapshot of the data item's most recent values or the state of the device at a point in time. It returns an MTConnectStreams XML document.

When the data streams between the initializer and the MTCAgent are transported by the Hypertext Transfer Protocol (HTTP), the MTCAgent works as an HTTP server and supports both types of commands, i.e., *Probe* and *Current* commands, and the initializer works as an anchor to connect the KRC to the physical machines and sends the requests as an HTTP client. As shown in Fig. 4(a), the initializer receives the extracted information of the machine's structure from the physical machines

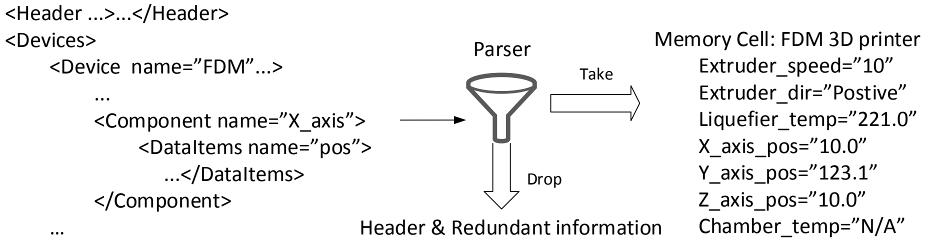


Fig. 5: The parser filters out component values and put into the Template.

after sending the *Probe* command to the MTCAgent. Fig. 4(b) shows the details of information received by the initializer after it sends the *Current* command to the MTCAgent to inquire the current status of the components in the physical machine.

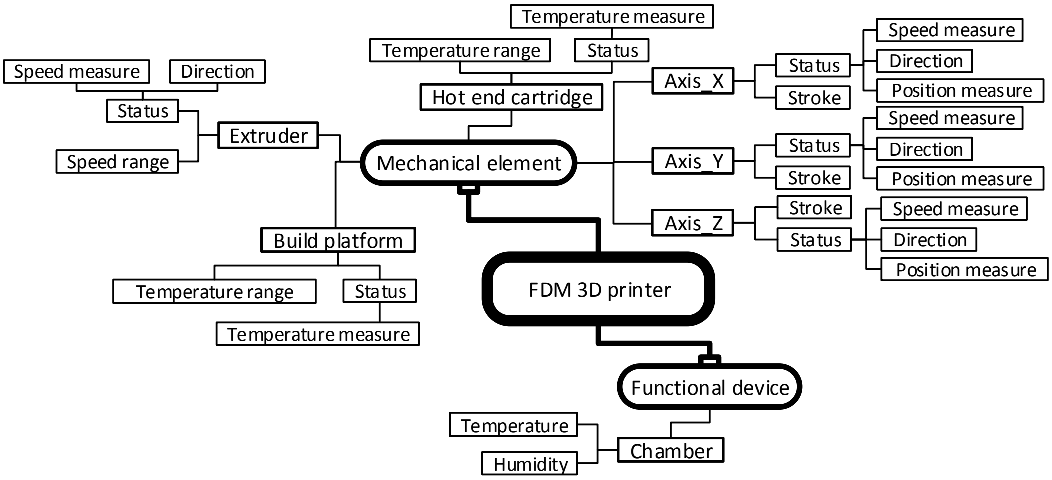
3.2.2. Parser

The data streams that get into the KRC through the initializer gateway cannot be easily digested without processing because the data are packed by XML structure [20] and encrypted by HTTP protocol standard [21]. A parser is used to help extract data from the complex XML structure. Note that the data in XML are organized by an architecture on the floor network level and have to be sequentially traversed to approach any specific portions. In addition, those packets brought by the HTTP protocol are embedded protocol's scripts, encryption, and identification code. These make the data stream from the initializer too complicated for direct data comprehension. To overcome these tangles, the parser provides a special mechanism for the data streams from the initializer to the data station in the KRC, shown in Fig. 5. Each data stream is passed by a service port in the parser, and a sequential traverse mechanism is immediately constituted to hop into the data structure. On the other hand, the processing center in the parser is ready to hone the raw data provided by the service port on the symmetrical processing mechanism. The symmetrical processing mechanism is conducted based on the simultaneous two-way handling of service port and processing center in the parser. The data are popped out from the service port and immediately put into and processed by the processing center. Because of this, the KRC can minimize the processing speed even at the first stages. In the latest steps in the parser processing, the data are quickly transferred to the data station for the next stages. All interactions of services in the KRC are coordinated by the configuration center, including the communication between the parser and the data station.

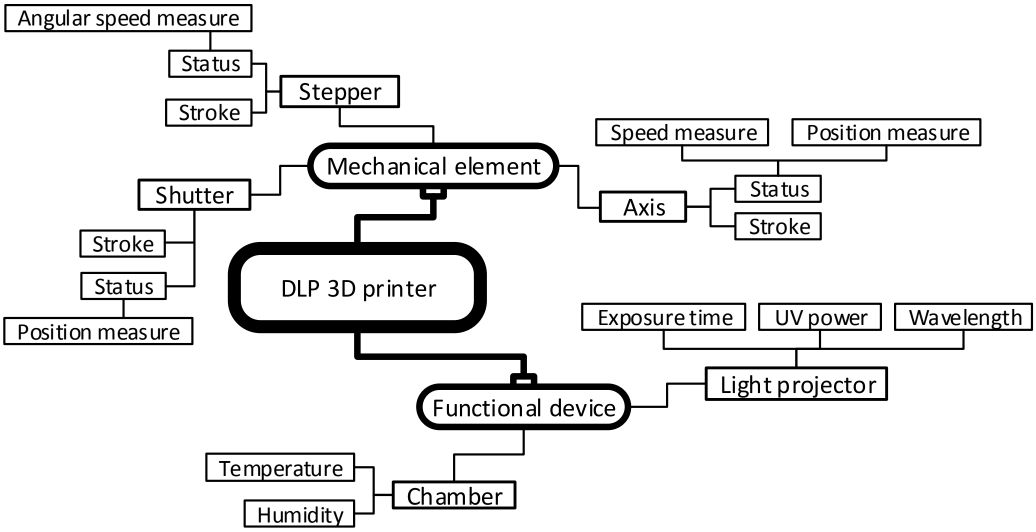
3.2.3. Data station and information model

A data station is built with a set of memory cells to store the data processed by the parser in the previous stage. It also works as a transit station to provide the data for the information model. The data include the current status of the physical machines such as a component's position, temperature, etc. These data are stored in the memory cells and controlled by the configuration center. These data sources form parts of the input for any digital twins and they determine all the parameters of the digital twin's components. By collaboration with the information model, the data station creates a closed working loop in which the information model provides information needed for digital twins and the data station supplies the needed values of parameters representing the physical machine's condition. As with the process of the parser, the collaboration is conducted on the symmetrical processing mechanism. Thus, a smooth work sequence is reached to prepare all necessary materials for building the digital twins. Because the way 3D printers handle materials and workpieces is different from CNC machines, in order to have a good representation of 3D printing machines, we propose an information model for building a type of digital 3D printers. In this model, we not only construct the mechanical elements of the 3D printer but also define the functional devices such as the laser, chamber, and electron beam.

Taking Fused Deposition Modeling (FDM) 3D printers as the example. This kind of 3D printer uses the X/Y/Z axes to carry the hot-end cartridge to move around the workspace, and it extrudes melted plastic filament layer by layer to produce a part. Figure 6(a) shows the mechanical elements and the functional devices of an FDM 3D printer with their parameters. The mechanical elements of an FDM 3D printer include the extruder, axes, hot-end cartridge, and build platform.



(a) Information model for FDM 3D printer.



(b) Information model for DLP 3D printer.

Fig. 6: Information models for FDM and DLP 3D printers.

The modeling of an axis is described by a set of variables, including stroke and status which includes direction, speed, and position. The stroke indicates the minimum and maximum range of each axis, and the status describes the current position, speed and direction of the axis movement. Each component in the functional device is constituted by subcomponents that describe the details of physical parts according to their functions. As shown in the FDM 3D printer's information model, there are parts working together to illustrate the chamber's temperature and humidity and constitute the functional device description, through which the configuration parameters of humidity and temperature of the FDM 3D printer are updated and controlled.

Figure 6(b) shows the information model for a Digital Light Processing (DLP) 3D printer. The DLP 3D

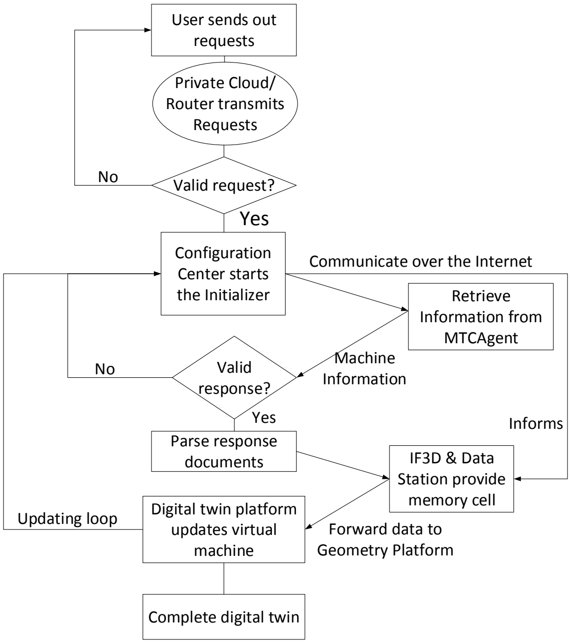


Fig. 7: The workflow of the proposed method.

printer uses a digital light projector to cure photopolymer resin into a solid 3D object one layer at a time. In this 3D printer, the angular stepper, axis and shutter are classified as the mechanical elements, which are responsible for individual functions and able to work independently. Likewise, the chamber and light projector constitute the functional device, in which the chamber is modeled by the temperature and humidity, and the light projector consists the exposure time, UV power and wavelength.

3.2.4. Configuration center and digital twin platform

The configuration center is indispensable in any smart manufacturing systems. The role of the configuration center is to coordinate hundreds of commands or interactions in the system. It is the brain of the cloud-based digital twin system and helps connect all parts as well as manage the logical processing in the KRC. The details of workflow of the configuration center are shown in Figure 7. When the user's requests come from a private cloud, the configuration center immediately activates a work sequence in the KRC. Starting from the initializer, the orders are received and then the initializer sends the communication requests to the MTAgent to inquire into the physical machines. As mentioned above, processing and exchanging data between the initializer and the parser are conducted based on the symmetrical processing mechanism where the processes operate at the same time. The key concept in this phase is that the configuration center helps synchronize the processes, times, and workloads between the initializer and the parser. In addition, the communication between the parser and the data station or the bridge between the data station and the information model is also coordinated by the configuration center. Thus, the configuration center is designed to provide a full knowledge of connecting all parts in the KRC and to minimize processing time in the whole system. In the terminal working sequence of the KRC system, the digital twins are presented at the digital twin platform.

4. Case Study and Evaluation

We study a system of two mini factories that are located at the Intelligent Systems Center of Missouri University of Science and Technology (MST mini-factory) and the University of Arkansas (UoA mini-factory). We use the Hardware-In-The-Loop (HIL) scheme to evaluate the cyber-physical manufacturing system. The system was developed and introduced in [22,23], which was used in the case study discussed below.

4.1. Case study

This section discusses the processes of building the digital twins for the two different types of 3D printers. The scenarios are designed to make the experiments close to the real manufacturing process. As shown in Figure 8, we use a Bukito 3D printer in the MST mini-factory, and a Ultimaker 3D printer in the UoA mini-factory. In the first scenario, a user sends a request to the private cloud server, and then it is coordinated to the KRC to request establishing the digital twin of the Bukito 3D printer. The request is first processed by the configuration center in the KRC. After the request validation is completed, the configuration center activates the initializer to contact the MST factory's MTCAgent of the first inquiry for the Bukito 3D printer information. The initializer forwards the MTCAgent response with an XML standard document to the parser under the supervision of the configuration center. Meanwhile, the configuration center notifies the data station to reserve places for the Bukito 3D printer information model that is transferred by the parser. The data including the physical machine's information are combined with the IF3D (information model of 3D printer) to completely describe the structure and status of the machine. Finally, the virtual duplicate of the physical machine is presented at the digital twin platform.

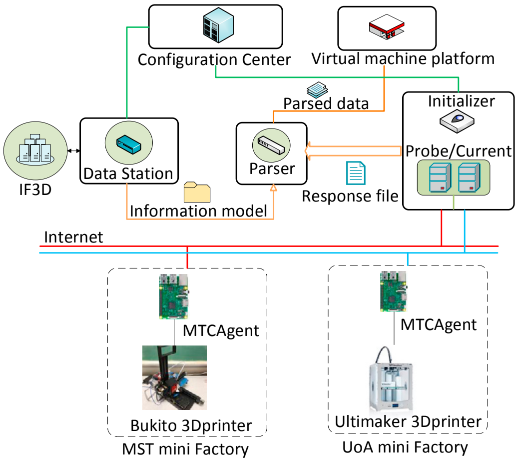
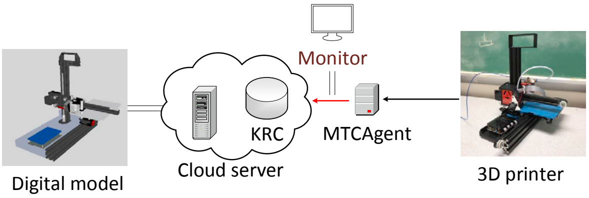


Fig. 8: Development of 3D printer digital twins for MST and UoA mini-factories.

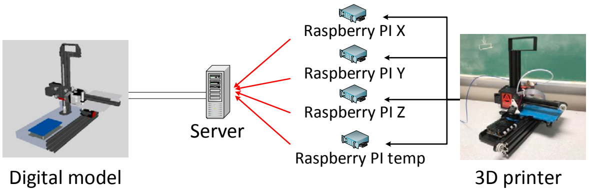
In the second scenario, the KRC receives two requests from different users coordinated by the private servers. One asks for constructing a digital twin for the Bukito 3D printer at the MST factory, and the other requests for constructing a digital twin for the Ultimaker 3D printer at the UoA factory. The requests are verified one by one by the configuration center in the KRC. Afterwards, the initializer is immediately activated to communicate with the two factories' MTCAgents by using Probe/Current commands to retrieve the information of machines. In the next steps, the initializer follows the same process as in the first scenario to forward the MTCAgent's responses to the parser. At this point, the configuration center notifies the data station to reserve two slots for the Bukito 3D printer's information model and Ultimaker 3D printer's information model. Since the Bukito 3D printer and the Ultimaker 3D printer use two different types of information models, the IF3D provides the types of models to combine with the corresponding information in the data stations.

4.2. Experimentation and evaluation

In this section, experiments are conducted on the testbed to evaluate the performance of the CBDT method. As shown in Figure 9, by using WireShark software [24], we can monitor the data streams in the system when the digital twins are being built. The capability of the CBDT method is evaluated, analyzed, and compared to the sensor-sever digital twin (SSDT) method [4,5,9]. In these experiments, the SSDT uses Raspberry PIs to monitor machine status, with each Raspberry PI monitoring one parameter of the machine.

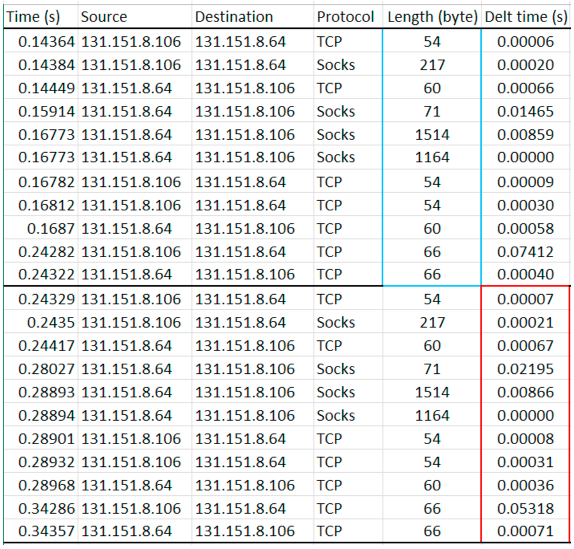


(a) Cloud-based digital twin (CBDT)



(b) Sensor-server digital twin (SSDT)

Fig. 9: Creation of digital twins by (a) CBDT and (b) SSDT for the physical 3D printer.



Time (s)	Source	Destination	Protocol	Length (byte)	Delt time (s)
0.14364	131.151.8.106	131.151.8.64	TCP	54	0.00006
0.14384	131.151.8.106	131.151.8.64	Socks	217	0.00020
0.14449	131.151.8.64	131.151.8.106	TCP	60	0.00066
0.15914	131.151.8.64	131.151.8.106	Socks	71	0.01465
0.16773	131.151.8.64	131.151.8.106	Socks	1514	0.00859
0.16773	131.151.8.64	131.151.8.106	Socks	1164	0.00000
0.16782	131.151.8.106	131.151.8.64	TCP	54	0.00009
0.16812	131.151.8.106	131.151.8.64	TCP	54	0.00030
0.1687	131.151.8.64	131.151.8.106	TCP	60	0.00058
0.24282	131.151.8.106	131.151.8.64	TCP	66	0.07412
0.24322	131.151.8.64	131.151.8.106	TCP	66	0.00040
0.24329	131.151.8.106	131.151.8.64	TCP	54	0.00007
0.2435	131.151.8.106	131.151.8.64	Socks	217	0.00021
0.24417	131.151.8.64	131.151.8.106	TCP	60	0.00067
0.28027	131.151.8.64	131.151.8.106	Socks	71	0.02195
0.28893	131.151.8.64	131.151.8.106	Socks	1514	0.00866
0.28894	131.151.8.64	131.151.8.106	Socks	1164	0.00000
0.28901	131.151.8.106	131.151.8.64	TCP	54	0.00008
0.28932	131.151.8.106	131.151.8.64	TCP	54	0.00031
0.28968	131.151.8.64	131.151.8.106	TCP	60	0.00036
0.34286	131.151.8.106	131.151.8.64	TCP	66	0.05318
0.34357	131.151.8.64	131.151.8.106	TCP	66	0.00071

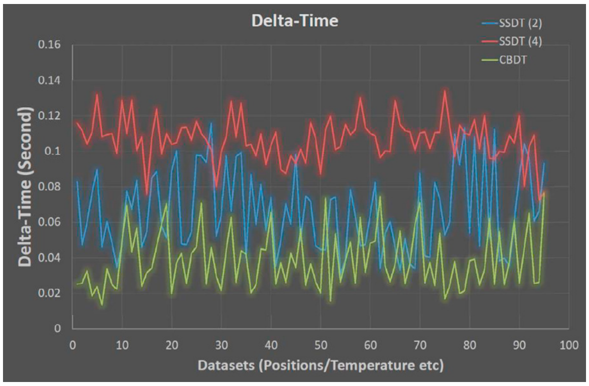
Fig. 10: Example of captured connections between MTCAgent and KRC.

Table 1: The delta-time of CBDT vs. SSDT.

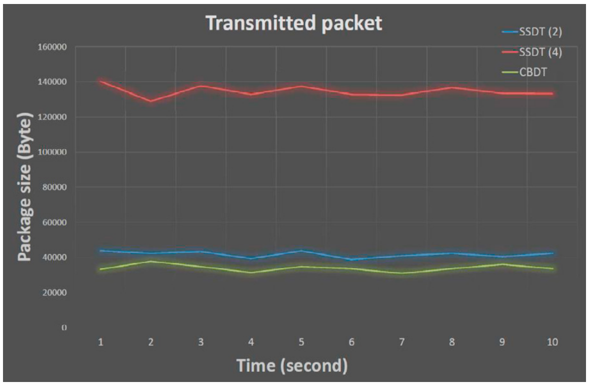
Data set (#)	SSDT(2)	SSDT(4)	CBDT
1	0.0830	0.1159	0.0252
2	0.0473	0.1117	0.0258
3	0.0601	0.1041	0.0327
4	0.0766	0.1110	0.0186
5	0.0900	0.1319	0.0240
6	0.0461	0.1083	0.0134
7	0.0607	0.1094	0.0341
8	0.0484	0.1101	0.0249
9	0.0347	0.0987	0.0223
10	0.0472	0.1288	0.0493

Table 2: The transmitted packet size of CBDT vs. SSDT.

Time(s)	SSDT(2)	SSDT(4)	CBDT
1	44015	140243	33468
2	42446	129275	37984
3	43553	137702	35030
4	39495	133037	31650
5	43944	137416	35030
6	38967	133006	33854
7	41090	132740	31159
8	42517	136706	31159
9	40633	133639	26260
10	42517	133400	33800



(a) Delta-time of CBDT vs. SSDT



(b) Packet size of CBDT vs. SSDT

Fig. 11: Delta-time and transmitted packet comparisons.

These Raspberry PIs are connected to the Internet and the server acquires information by sending commands to each of them. As for CBDT, the MTCAgent is responsible for providing machine information for building the digital twin over the Internet.

The Bukito printer at the MST mini-factory is adapted to support tasks with the MTConnect protocol. A printing task which takes 10 s to finish is assigned to the printer. During the process of building digital twins, the digital twin platform sends requests to the configuration center with the frequency of 10 Hz to inquire and update the machine's status. The performance of the KRC is measured based on predefined parameters as follows:

- Delta-time: the time required of the KRC to receive one dataset over the Internet from the physical machine.
- Packet size: the size of the dataset and courier packet transmitted between the server and the local factory.

For the two parameters, the delta-time is estimated whenever the physical machine updates its status and the size of packet transmitted is calculated per unit time. Note that the delta-time does not include the processing time in the cloud/server. We set up a mechanism using the WireShark between the MTCAgent and the KRC to measure these values. Figure 10 shows the data streams between the MTCAgent and the KRC. In every communication, the KRC works as an HTTP client with the IP address 131.151.8.106 and sends the request through the initializer to the HTTP server (MTCAgent) with the IP address 131.151.8.64. The information from line 2 to line 12 in Figure 10 shows the values of the machine's parameters provided by the MTCAgent. The numbers in the red box are the consuming time during the transmission of data from the source IP address to the destination IP address. We can estimate the total delta-time of each stage by summing the delta-time of each transmission within the stage. The numbers in the blue box show the exchanged packet size of each data transferring, and the total transmitted packet size is calculated by summing the packet size in this section.

To evaluate the efficiency of the proposed method for building cloud-based digital twins, a method of data transmission that is extended from the sensor-server digital twin (SSDT) method [4,5,9] is presented here in two cases, the SSDT (2) and the SSDT (4), to compare with our method in terms of the delta-time and the packet size. Raspberry PIs are connectted to the Bukito 3D printer to read the status and send the data over the Internet to the server. The CBDT aggregates all four parameters (X/Y/Z/temperature) from the 3D printer. In the SSDT (2), two Raspberry PIs are used to monitor the printer head's z-axis position and y-axis position. The SSDT (4) uses four Raspberry PIs to monitor the physical machine including X/Y/Z positions plus temperature.

From the WireShark, we can monitor the data streams between the MTCAgent/sensors and the server. To have a more intuitive view of the performance, we calculate the delta-time of each method. We get the delta-time and transmitted packet size, as shown in Table 1 and Table 2. Figure 11 shows the comparison of the CBDT, SSDT (2), and SSDT (4) in terms of the delta-time for each dataset. From the figure we can see that SSDT (4) and SSDT (2) use more time to transmit datasets to the server than the CBDT method. This is because SSDT (2) and SSDT (4) use Raspberry PIs that directly send the data to the server. When using Raspberry PIs, the server is only able to receive one set of data at a time, thus it takes more time compared to the CBDT which uses MTCAgent to send multiple sets of data at a time. The larger the number of deployed sensors in the SSDT, the longer the delta-time required.

Figure 11(b) plots the transmitted packet size in unit time using Table 2. Note that the CBDT and the SSDT send the same amount of valid data to the server, but SSDT (2) and SSDT (4) require larger total packet sizes. It is clear that the CBDT has better performance than SSDT (2) and SSDT (4) as the CBDT requires a smaller total packet size.

5. Conclusion

In this paper, we have developed a method of developing digital twins in a CPCM system using an information model and MTConnect protocol. The main contributions of this paper are summarized as follows: 1) the concept and architecture of the Knowledge Resource Center (KRC) are developed; 2) the functionalities and mechanisms of KRC components are defined; and 3) the proposed method has been successfully evaluated by implementing it in a CPCM testbed. By taking advantage of the KRC, the cloud-based digital twins successfully reduce the system's overhead and provide an effective CPCM application.

Acknowledgements

This project is supported by NSF Grant CMMI 1551448 entitled "EAGER/Cybermanufacturing: Architecture and Protocols for Scalable Cyber-Physical Manufacturing Systems".

References

[1] Zhang, L., Luo, Y., Tao, F., Li, B.H., Ren, L., Zhang, X., et al. Cloud manufacturing a new manufacturing paradigm. Enterprise Information Systems 2014;8(2):167–187. doi:`10.1080/17517575.2012.683812`.

[2] Hanwu, H., Yueming, W.. Web-based virtual operating of {CNC} milling machine tools. Computers in Industry 2009;60(9):686 – 697. doi:`https://doi.org/10.1016/j.compind.2009.05.009`.

[3] Wang, L., Trngren, M., Onori, M.. Current status and advancement of cyber-physical systems in manufacturing. Journal of Manufacturing Systems 2015;37(Part 2):517 – 527. URL: `http://www.sciencedirect.com/science/article/pii/S0278612515000400`. doi:`https://doi.org/10.1016/j.jmsy.2015.04.008`.

[4] Konstantinov, S., Ahmad, M., Ananthanarayan, K., Harrison, R.. The cyber-physical e-machine manufacturing system: Virtual engineering for complete lifecycle support. Procedia CIRP 2017;63(Supplement C):119 – 124. doi:`https://doi.org/10.1016/j.procir.2017.02.035`; manufacturing Systems 4.0 Proceedings of the 50th CIRP Conference on Manufacturing Systems.

[5] Tao, F., Zhang, M.. Digital twin shop-floor: A new shop-floor paradigm towards smart manufacturing. IEEE Access 2017;5:20418–20427. doi:`10.1109/ACCESS.2017.2756069`.

[6] Rosen, R., von Wichert, G., Lo, G., Bettenhausen, K.D.. About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine 2015;48(3):567 – 572. doi:`https://doi.org/10.1016/j.ifacol.2015.06.141`; 15th IFAC Symposium onInformation Control Problems inManufacturing.

[7] Schroeder, G.N., Steinmetz, C., Pereira, C.E., Espindola, D.B.. Digital twin data modeling with automationml and a communication methodology for data exchange. IFAC-PapersOnLine 2016;49(30):12 – 17. doi:`https://doi.org/10.1016/j.ifacol.2016.11.115`; 4th IFAC Symposium on Telematics Applications TA 2016.

[8] Angrish, A.. Development and evaluation of data models and system architectures for virtual manufacturing machines. North Carolina State University 2017;URL: `https://repository.lib.ncsu.edu/bitstream/handle/1840.20/34725/etd.pdf?sequence=1`.

[9] Wang, L., Gao, R., Ragai, I.. An integrated cyber-physical system for cloud manufacturing. ASME Paper No MSEC2014-4171 2014;.

[10] Wang, X., Chen, X., Yuen, C., Wu, W., Zhang, M., Zhan, C.. Delay-cost tradeoff for virtual machine migration in cloud data centers. Journal of Network and Computer Applications 2017;78:62 – 72. doi:`https://doi.org/10.1016/j.jnca.2016.11.003`.

[11] MTConnect Protocol. 2008. URL: `http://www.mtconnect.org/`.

[12] Michaloski, J., Lee, B., Proctor, F., Venkatesh, S., Ly, S.. Quantifying the performance of mt-connect in a distributed manufacturing environment. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 2009;2:533 – 539. doi:`10.1115/DETC2009-86666`; 29th Computers and Information in Engineering Conference, Parts A and B.

[13] Lee, B.E., Michaloski, J., Proctor, F., Venkatesh, S., Bengtsson, N.. Mtconnect-based kaizen for machine tool processes. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 2010;3:1183 – 1190. doi:`10.1115/DETC2010-28763`; 30th Computers and Information in Engineering Conference, Parts A and B.

[14] Vijayaraghavan, A., Sobel, W., Fox, A., Dornfeld, D., Warndorf, P.. Improving machine tool interoperability using standard-

ized interface protocols: Mt connect. Laboratory for Manufacturing and Sustainability 2008;.

[15] Xu, X.. From cloud computing to cloud manufacturing. Robotics and Computer-Integrated Manufacturing 2012;28(1):75 – 86. doi:`https://doi.org/10.1016/j.rcim.2011.07.002`.

[16] Diaz, N., Helu, M., Jarvis, A., Tönissen, S., Dornfeld, D., Schlosser, R.. Strategies for minimum energy operation for precision machining. Laboratory for Manufacturing and Sustainability 2009;.

[17] Lee, J., Bagheri, B., Kao, H.A.. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing Letters 2015;3(Supplement C):18 – 23. URL: `http://www.sciencedirect.com/science/article/pii/S221384631400025X`. doi:`https://doi.org/10.1016/j.mfglet.2014.12.001`.

[18] Wu, D., Greer, M.J., Rosen, D.W., Schaefer, D.. Cloud manufacturing: Strategic vision and state-of-the-art. Journal of Manufacturing Systems 2013;32(4):564 – 579. URL: `http://www.sciencedirect.com/science/article/pii/S0278612513000411`. doi:`https://doi.org/10.1016/j.jmsy.2013.04.008`.

[19] Wu, D., Rosen, D.W., Wang, L., Schaefer, D.. Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. Computer-Aided Design 2015;59:1 – 14. URL: `http://www.sciencedirect.com/science/article/pii/S0010448514001560`. doi:`https://doi.org/10.1016/j.cad.2014.07.006`.

[20] XML Protocol. 2016. URL: `https://www.w3.org/XML/`.

[21] HTTP Protocol. 2014. URL: `https://www.w3.org/Protocols/`.

[22] Liu, X.F., Shahriar, M.R., Sunny, S.N.A., Leu, M.C., Hu, L.. Cyber-physical manufacturing cloud: Architecture, virtualization, communication, and testbed. Journal of Manufacturing Systems 2017;43:352 – 364. doi:`http://dx.doi.org/10.1016/j.jmsy.2017.04.004`; high Performance Computing and Data Analytics for Cyber Manufacturing.

[23] Nguyen, N.T., Leu, M.C., Liu, X.F.. Real-time communication for manufacturing cyber-physical systems. In: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA). 2017, p. 1–4. doi:`10.1109/NCA.2017.8171361`.

[24] Wireshark. 2017. Https://www.wireshark.org/.