7-2020

Towards a Cyber-Physical Manufacturing Cloud through Operable Digital Twins and Virtual Production Lines

Md Rakib Shahriar
University of Arkansas, Fayetteville

Towards a Cyber-Physical Manufacturing Cloud through Operable Digital Twins and Virtual Production Lines

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Science

by

Md Rakib Shahriar
Bangladesh University of Engineering & Technology
Bachelor of Science in Computer Science & Engineering, 2009

July 2020
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Xiaoqing "Frank" Liu, Ph.D.
Dissertation Director

Brajendra Nath Panda, Ph.D.
Committee Member

Miaoqing Huang, Ph.D.
Committee Member

Qinghua Li, Ph.D.
Committee Member

Wenchao Zhou, Ph.D.
Committee Member

ABSTRACT

In last decade, the paradigm of Cyber-Physical Systems (CPS) has integrated industrial manufacturing systems with Cloud Computing technologies for Cloud Manufacturing. Up to 2015, there were many CPS-based manufacturing systems that collected real-time machining data to perform remote monitoring, prognostics and health management, and predictive maintenance. However, these CPS-integrated and network ready machines were not directly connected to the elements of Cloud Manufacturing and required human-in-the-loop. Addressing this gap, we introduced a new paradigm of Cyber-Physical Manufacturing Cloud (CPMC) that bridges a gap between physical machines and virtual space in 2017. CPMC virtualizes machine tools in cloud through web services for direct monitoring and operations through Internet. Fundamentally, CPMC differs with contemporary modern manufacturing paradigms. For instance, CPMC virtualizes machining tools in cloud using remote services and establish direct Internet-based communication, which is overlooked in existing Cloud Manufacturing systems. Another contemporary, namely cyber-physical production systems enable networked access to machining tools. Nevertheless, CPMC virtualizes manufacturing resources in cloud and monitor and operate them over the Internet. This dissertation defines the fundamental concepts of CPMC and expands its horizon in different aspects of cloud-based virtual manufacturing such as Digital Twins and Virtual Production Lines.

Digital Twin (DT) is another evolving concept since 2002 that creates *as-is* replicas of machining tools in cyber space. Up to 2018, many researchers proposed state-of-the-art DTs, which only focused on monitoring production lifecycle management through simulations and data driven analytics. But they overlooked executing manufacturing processes through DTs from virtual space. This dissertation identifies that DTs can be made more productive if they engage

directly in direct execution of manufacturing operations besides monitoring. Towards this novel approach, this dissertation proposes a new operable DT model of CPMC that inherits the features of direct monitoring and operations from cloud. This research envisages and opens the door for future manufacturing systems where resources are developed as cloud-based DTs for remote and distributed manufacturing. Proposed concepts and visions of DTs have spawned the following fundamental researches.

This dissertation proposes a novel concept of DT based Virtual Production Lines (VPL) in CPMC in 2019. It presents a design of a service-oriented architecture of DTs that virtualizes physical manufacturing resources in CPMC. Proposed DT architecture offers a more compact and integral service-oriented virtual representations of manufacturing resources. To re-configure a VPL, one requirement is to establish DT-to-DT collaborations in manufacturing clouds, which replicates to concurrent resource-to-resource collaborations in shop floors. Satisfying the above requirements, this research designs a novel framework to easily re-configure, monitor and operate VPLs using DTs of CPMC.

CPMC publishes individual web services for machining tools, which is a traditional approach in the domain of service computing. But this approach overcrowds service registry databases. This dissertation introduces a novel fundamental service publication and discovery approach in 2020, OpenDT, which publishes DTs with collections of services. Experimental results show easier discovery and remote access of DTs while re-configuring VPLs. Proposed researches in this dissertation have received numerous citations both from industry and academia, clearly proving impacts of research contributions.

ACKNOWLEDGEMENTS

Primarily, I would like to express my sincere gratitude to my doctoral supervisor Dr. Xiaoqing “Frank” Liu. He has provided me invaluable expertise and guidance throughout my tenure as a Ph.D. student. I am also thankful for his consistent professional and personal support. Without his supervision, this work would not have materialized. He is a man from whom one can learn perseverance.

In addition, I am thankful to my co-workers, particularly S M Nahian Al Sunny for his consistent support to brainstorm and implement the ideas. I am also thankful to my other colleagues such as Md Mahfuzer Rahman Limon, Joseph Sirriaanni, and Najla Althuniyan for keeping a research friendly and peaceful lab environment.

I would also like to mention that this dissertation was partially supported by NSF Grant CMMI 1551448 entitled "EAGER/Cybermanufacturing: Architecture and Protocols for Scalable Cyber-Physical Manufacturing Systems".

DEDICATION

I would like to dedicate this dissertation to

my mother Sultana Fatema,

my father Md Rafiqul Islam,

and

my beautiful wife Tarana Tasmin Bipasha.

TABLE OF CONTENTS

CHAPTER 1: CYBER-PHYSICAL MANUFACTURING CLOUD (CPMC)................... 1

1.1 Introduction.. 1

1.2 Literature Survey 4

1.3 Scalable and service-oriented layered architecture of CPMC 9

1.3.1 Scalable service-oriented architecture of CPMC.. 11

1.4 Virtualization of manufacturing resources using RESTful web services 13

1.5 Publication and Subscription of Manufacturing Applications and Web Services 16

1.5.1 Application and web service publishing center of the CPMC.................................... 16

1.5.2 Workflow: Manufacturers, Application Developers, and Consumers.......................... 16

1.6 Testbed implementation.. 17

1.6.1 Infrastructure implementations 18

1.6.2 Evaluation of the testbed in manufacturing application scenarios............................... 22

1.6.3 Result Discussion... 25

1.6.4 State-of-the-art comparison 28

1.7 Conclusion 28

CHAPTER 2: VIRTUALIZING PHYSICAL MACHINE OPERATIONS INTO THE DIGITAL TWINS IN CYBER-PHYSICAL MANUFACTURING CLOUD 30

2.1 Introduction.. 30

2.2 Literature Survey 33

2.2.1 Service-oriented Virtualization and Communication methods of physical machine tools 33

2.2.2 Virtualization of physical machines through Digital Twins.. 35

2.3 Method of virtualizing manufacturing operations using MTComm in CPMC................ 36

2.3.1 Virtualization of manufacturing operations using MTComm 37

2.3.2 Virtualization of manufacturing operation information of machining tools in CPMC 37

2.4 Integration of virtualized manufacturing operations into 3D visual digital twin in CPMC 38

2.4.1 Granular level assembling of 3D models to form digital twins using MTComm ontology 38

2.4.2 Incorporating manufacturing operations into 3D visual digital-twin components 39

2.5 Evaluation using manufacturing application scenarios.. 40

2.5.1 Application Scenario 1: Components movement simulation during manufacturing operation in the 3D visual digital twin of Bukito 3D printer.. 41

2.5.2 Application Scenario 2: Low filament notification by symbolic change in 3D visual digital twin of Bukito 3D printer 41

2.5.3 Application Scenario 3: Ensuring maintenance safety by changing colors of heated components in a Digital-Twin 43

2.5.4 Application Scenario 4: Move axes of Bukito machine from the Digital-Twin........... 44

2.5.5 Application Scenario 5: Operate electrical power supply of Bukito machine from the Digital-Twin.. 45

2.6 Results discussion 45

2.7 Delay analysis of 3D rendering during runtime of Digital Twins in CPMC 45

2.7.1 Digital Footprint creation.. 47

2.8 Conclusion 47

CHAPTER 3: VIRTUAL PRODUCTION LINES USING DIGITAL TWINS IN CYBER-PHYSICAL MANUFACTURING CLOUD 49

3.1 Introduction 49

3.2 Literature Survey 53

3.2.1 Virtual Production Lines in Cloud Manufacturing 54

3.2.2 Virtualizing characteristics and capabilities of manufacturing resources as Digital Twins 55

3.2.3 Cloud-based virtual manufacturing collaborations 57

3.3 Research issues and objectives 58

3.4 Service-Oriented Architecture of Digital Twins for CPMC 61

3.4.1 Architecture of DTs 61

3.4.2 Virtualizing manufacturing resources in manufacturing clouds through DTs............ 64

3.4.3 DT-to-DT collaboration in CPMC 68

3.5 Framework for DT-based Virtual Production Lines in CPMC 71

3.5.1 Conceptualizing reconfigurable VPLs in CPMC 72

3.5.2 Framework for configuring, monitoring and operating DT-based VPLs in CPMC 73

3.6 Testbed implementation and case validations 79

3.6.1 Testbed infrastructure design 80

3.6.2 Case validations 85

3.6.3 Discussion 88

3.7 Conclusion 91

CHAPTER 4: A FRAMEWORK FOR PUBLICATION AND DISCOVERY OF DIGITAL TWINS 93

4.1 Introduction 93

4.2 Motivation 96

4.2.1 Motivating Scenario 96

4.2.2 Need for a novel product-oriented publication approach 98

4.2.3 Research Challenges 99

4.3 Reference framework of OpenDT 100

4.3.1 Framework design 101

4.3.2 Service-oriented architecture of DTs 103

4.3.3 Service Collection Descriptor model 107

4.3.4 DT Query model 108

4.4 Implementation 109

4.4.1 Results discussion 112

4.5 Conclusion 115

REFERENCES 117

ALL PUBLISHED AND SUBMITTED PUBLICATIONS 125

LIST OF TABLES

Table 1: Feature wise comparison between this work and other state-of-the-arts to contrast the key conceptual differences. 89

Table 2: List of evaluation metrics 112

LIST OF FIGURES

Figure 1: Conceptual model of Cyber-Physical Manufacturing Cloud (CPMC) 9

Figure 2: A four-layer architecture of CPMC. 10

Figure 3: Workflow of Virtualization method in CPMC 14

Figure 4: Virtualization data of a manufacturing resource using web service for 3D printing in JSON 15

Figure 5: Application and services for manufacturing operations development hierarchy 16

Figure 6: Workflow of the participants in the CPMC system 17

Figure 7: Testbed implementation of CPMC 18

Figure 8: Subscribe applications in the Application Center 20

Figure 9: Application Center dashboard showing subscribed applications 20

Figure 10: Dashboard and Monitoring Operation view of a third-party smartphone application 21

Figure 11: Service-Publishing Center – Publish a web service 21

Figure 12: Search Engine of published web services 22

Figure 13: Use Case diagram for manufacturing application scenario 1 22

Figure 14: Sequence Diagram for manufacturing application scenario 1 23

Figure 15: Use Case diagram of manufacturing application scenario 2 24

Figure 16: Sequence Diagram for manufacturing application scenario 2 24

Figure 17: Response time of MTConnect services for monitoring operations 27

Figure 18: Method of virtualizing a manufacturing resource using MTComm [48] 35

Figure 19: Mapping of manufacturing operations on machine level and machine 36

Figure 20: Mapping of real-time manufacturing operation status data streaming 37

Figure 21: A breakdown of a Bukito 3D printer machine following MTComm based ontology information 39

Figure 22: A component level sequence diagram illustrating the process of digital-twin construction. 40

Figure 23: 3D visualized digital twin of Bukito 3D printer rendered on an iPad application 41

Figure 24: (a) Filament reel in Digital-Twin changes its color to red while weight of the filament reel is below a threshold value; (b) Load Cell setup to measure reel weight; (c) Amplifier for measuring weight of the filament reel. 42

Figure 25: The hot end carriage of the Bukito machine is changed to Red when its temperature is above 200°F degrees Fahrenheit. 43

Figure 26: MTComm standard operation XML that sets X-Axis value to 100 of Bukito machine. 44

Figure 27: IoT Relay is utilized to control electrical power of the Bukito machine through an MTComm operation service. 45

Figure 28: Graph showing network delays on rendering machine movements into Digital-Twin 3D visualization. Red line in indicates a busy machine while green indicates the opposite. 46

Figure 29: A service-oriented architecture of DT for CPMC. This architecture virtualizes a manufacturing resource and creates a synchronous representation using data driven and visualization driven monitoring. The Cloud Twin executes physical machine operations directly and capable of establishing DT-to-DT collaboration in cloud within a virtual manufacturing process.. 62

Figure 30: Five steps of virtualization method to construct a DT in CPMC. 65

Figure 31: Semantic ontology structure of an MTComm compliant manufacturing resources. .. 66

Figure 32: An example virtualization information mapping for a Bukito 3D printer for monitoring and operations in a DT. 67

Figure 33: DT-to-DT Collaboration in CPMC. (a) presents web services offered by a DT to establish collaboration; (b) presents a manufacturing task information model for executing a manufacturing operation and invoke another DT for event-driven collaborations; (c) presents a collaboration information model to validate collaboration between two DTs. 68

Figure 34: An example workflow showing a CNC DT collaborating with a robotic arm DT in CPMC. This workflow also details how DTs take decision on next task (Figure 7 (b)) in an event driven collaborative manufacturing process. 70

Figure 35: Conceptual diagram of DT-based VPL in CPMC. Each VPL associates a virtual manufacturing process of different configurations showcase high-flexibility and reconfigurability production lines.. 72

Figure 36: Framework design of DT-based VPL in CPMC. This framework has four layers as in CPMC, except the functional components are different. This framework employs HTTP based communications in all layers. 74

Figure 37: Swimlane diagram to present DT-based VPL configuration workflow in CPMC. Generated configurations can support different manufacturing processes such as discrete, event-driven, collaborative, sequential, parallel etc. 76

Figure 38: VPL configuration data model for different manufacturing processes. This model refers back to Task model presented in Figure 33 (b). 77

Figure 39: Process execution overflow of DT-based VPL in CPMC where cloud-based DTs operate MR based tasks. This execution workflow supports event-driven collaborative and discrete-parallel manufacturing processes. 79

Figure 40: Testbed infrastructure of DT-based VPLs in CPMC that is reconfigurable. There are five machines arranged in two physical shop floors. Each of these machines are virtualized as DTs in the University of Arkansas private cloud. Each DTs are hosted in separate Linux virtual machine instances in the University of Arkansas private cloud. Multiple web applications are developed in a cloud server, which access these DTs to configure, concurrently simulate and remotely operate VPL. 80

Figure 41: (a) Overall communication methods in the testbed. (b) Virtualizing an Ultimaker2 machine as a DT in CPMC using MTComm. 81

Figure 42: (a) A state diagram to describe extraction of virtualization information from MTComm probe document in a DT. (b) A flow chart to generate MTComm's Operation Request message. 83

Figure 43: Bottom-up aggregation of 3D visualization in DTs from the testbed. (a), (b), (c), and (d) presents aggregation of Ultimaker2, X-Carve CNC, Bukito, and Uarm respectively. 84

Figure 44: Collaboration information for all five implemented DTs. This information is applied while configuring a VPL. 85

Figure 45: A DT-based VPL in CPMC consisting of two DTs from two shop floors. 86

Figure 46: A VPL consisting of three DTs and they are all from Shop Floor 2. 87

Figure 47:A sequence diagram to describe DT-to-DT virtual collaborations in CASE VALIDATION 2. 87

Figure 48: A motivating scenario to describe the new publication approach based on the idea of product catalogs. 96

Figure 49: A high-level view on OpenDT framework. This diagram presents basic operations of OpenDT such as DT publication, registry, and discovery. 100

Figure 50: A service-oriented architecture of DT. Cyber-Physical Communication is applicable in the contexts of CPS and IoT. 104

Figure 51: Class diagram of Digital Twin interface. This interface includes general attributes and methods of a DT. DT developers define implementation of this interface for use of applications and mashups.......... 106

Figure 52: Information model of DT descriptor. 108

Figure 53: Query data hierarchy in OpenDT. 109

Figure 54: Testbed structure for case validations. All five machines are represented by four DTs. They collaborate in two groups such as Composite-1 (Bukito-Uarm1) and Composite-2 (Bukito-Uarm2-XCarve). Decision support during collaborations are established from the DTs. Established collaborations are event-driven and service-oriented. 110

Figure 55: Testbed Results. (a) presents Line of Code (LOC) reduction in SPAs. (b) depicts reduction of service registries using OpenDT. 113

CHAPTER 1: CYBER-PHYSICAL MANUFACTURING CLOUD (CPMC)

Abstract - Cyber-physical systems are integrations of computation, networking, and physical processes and they are increasingly finding applications in manufacturing. Cloud manufacturing integrates cloud computing and service-oriented technologies with manufacturing processes and provides manufacturing services in manufacturing clouds. A cyber physical system for manufacturing is not a manufacturing cloud if it does not use virtualization technique in cloud computing and service-oriented architecture in service computing. On the other hand, a manufacturing cloud is not cyber physical system if it does not have components for direct interactions with machine tools and other physical devices. In this chapter, a new paradigm of Cyber-Physical Manufacturing Cloud (CPMC) is introduced and described to bridge gaps among cloud computing, cyber physical systems, and manufacturing. A CPMC allows direct operations and monitoring of machine tools in a manufacturing cloud over the Internet. A scalable and service-oriented layered architecture of CPMC is developed. It allows publication and subscription of manufacturing web services and cross-platform applications in CPMC. A virtualization method of manufacturing resources in CPMC is presented. In addition, communication mechanisms between the layers of the CPMC using communication protocols such as MTConnect, TCP/IP, and REST are discussed. A CPMC testbed is developed and implemented based on the proposed architecture. The testbed is fully operational in two geographically distributed sites. The developed testbed is evaluated using several manufacturing scenarios. Its testing results demonstrate that it can monitor and execute manufacturing operations remotely over the Internet efficiently in a manufacturing cloud.

1.1 Introduction

Cloud manufacturing (CMfg) is an integration of cloud and service computing with manufacturing processes [1, 2] . The number of cloud manufacturing systems is increased due to the rapid research and development of CMfg. Cyber Physical Systems (CPS) are another paradigm, which integrates computation, networking, and physical processes and allows directly operating and monitoring physical devices over the Internet and finds applications in manufacturing too [3, 4]. Hence, it is necessary to integrate cloud manufacturing and cyber physical systems for providing manufacturing services, which can directly operate and monitor machine tools in a manufacturing cloud. As a result, a new paradigm of Cyber Physical Manufacturing Clouds (CPMC) is introduced. A CPMC is a type of manufacturing clouds where machining tools can be directly monitored and operated over the Internet from clouds. Many existing manufacturing clouds provide manufacturing services and but do not allow their machining tools to be monitored and operated directly from the clouds. For example, Shapeways provide human-in-the-loop cloud manufacturing services. They allow customers to order 3D printing services from a cloud, human operators manually operate 3D machines to perform manufacturing processes. It is desirable to further automate manufacturing processes in manufacturing clouds by directly operating machining tools and monitoring their manufacturing processes with less human interventions. On the other hand, several cyber physical systems have been developed for manufacturing [5, 6]. They allow direct operation and monitoring of machine tools over the Internet. However, they are not manufacturing clouds with any virtualization methods and are not based on service-oriented architecture.

Several interesting research issues in CPMC need to be addressed. Firstly, a scalable service-oriented architecture needs to be developed for CPMC. Secondly, virtualization of manufacturing resources needs to be investigated. Thirdly, communication mechanisms are needed to enable network communications among components of CPMC. There are many networks enabled manufacturing tools in the existing manufacturing environments, but very few tools are utilized due to the lack of cyber-physical manufacturing clouds. Application scope of CPMC is widened by the recent developments of Industry 4.0 [6]. Therefore, the potential use of CPMC in manufacturing processes can significantly improve the use of underutilized network enabled manufacturing machines.

We present a scalable service-oriented architecture of CPMC. It allows manufacturers to publish and subscribe web services for manufacturing operations and manufacturing applications to monitor and execute manufacturing operations remotely over the Internet. It also creates an open platform for the manufacturers and customers to develop rich Internet applications. Virtualization of manufacturing resources in CPMC is important due to the diversity and complexity of the manufacturing resources. The presented CPMC system virtualizes manufacturing tools by representing functionalities of machining tools using web services and other machining tool characteristics. Communications between layers of the layered architecture are critical due to the diversity of the technical characteristics of each layer. The communication mechanisms are developed using MTConnect, Transmission Control Protocol/Internet Protocol (TCP/IP), and REpresentational State Transfer (REST) protocol.

An operational testbed is implemented based on the layered architecture of CPMC. The testbed consists of two groups of manufacturing machines located in geographically distributed sites – the University of Arkansas (UARK), and the Missouri Institute of Science and Technology (MST). The testbed cloud is hosted in the network of UARK. These manufacturing machines are virtualized as REpresentational State Transfer based (or RESTful) web services and published using the service-publishing center in the cloud. Multiple manufacturing applications are developed using the published web services over platforms such as Web, iOS, and Android. An application center is developed to publish manufacturing applications. These published applications are ready to be subscribed and used for manufacturing. MTConnect protocol is used in the testbed to monitor manufacturing machines remotely over the Internet. MTConnect is a popular RESTful Internet communication protocol for collecting status information of the machine tools. By design, it supports read-only communication of the machines. Consequently, Transmission Control Protocol/Internet Protocol (TCP/IP) is used in order to operate the manufacturing machines.

Our contributions in this work are:

i. Design of a scalable and service-oriented layered architecture of CPMC
ii. Development of a virtualization method of manufacturing resources for CPMC
iii. Development of communication mechanisms for components of CPMC
iv. Implementation of a fully operational testbed of CPMC based on the architecture

1.2 Literature Survey

Cloud manufacturing is an emerging manufacturing paradigm. The concept of cloud manufacturing initially was introduced in 2010 [1]. Several researches have been conducted by them on the architecture of cloud manufacturing [7, 8, 9, 10]. According to Esmaeilian et al, research in cloud manufacturing can be classified into three categories: studies focused on designing platform and data-sharing architecture, studies concentrated on resource allocation and management, and studies aimed at describing new business models and cloud manufacturing applications. The first group of studies describes the Information Technology (IT) systems requirement and data integration platforms [11]. Huang et al. [12] discussed an architecture of cloud manufacturing service platforms in small and medium-sized enterprises. Suo and Gao [13] illustrated how simulation software packages can be used to find the optimal configuration for cloud manufacturing service platforms. Zhang et al. [14] demonstrated an application of fiber optic sensing in improving the efficiency of cloud manufacturing service platforms. The paradigm of cloud manufacturing enables companies to expand their capabilities and have flexibility in scale depending on market demand. Helo et al. [15] described a limitation of current IT solutions in distributed manufacturing in a multi-layer supply chain and proposed a new cloud-based centralized software application, where a proper information-sharing infrastructure is provided between suppliers and other supply chain entities. Tao et el [7] proposed a computing and service-oriented manufacturing model describing a ten-layer cloud-manufacturing architecture incorporating the regular manufacturing processes and applications. The architecture also provided important insights on the manufacturing machine virtualization. However, method of virtualizing the manufacturing machines were not discussed elaborately. In addition, the context of communication between layers were not described in the architecture description. Several other studies on cloud manufacturing were conducted. A high-level four-layer architecture of cloud manufacturing was proposed by Xu [2]. Numerous researchers provided surveys of cloud manufacturing [16, 17, 10, 18]. A number of prototypes of cloud manufacturing were developed [19, 20]. The above research efforts projects some progresses on scalable architecture of cloud manufacturing. However, the aforementioned researches do not address the unique needs of cyber-physical manufacturing clouds where manufacturing machines are operated and monitored directly from clouds based on an integration of the internet technologies, deeply embedded computing, automatic control and monitoring, and networked manufacturing.

The paradigm of cyber-physical systems has gained momentum in research and development [21, 13]. Recently cyber-physical systems are finding applications in manufacturing

[21, 10, 22]. A cyber-physical production system was introduced [22]. Wang et al. [18] presented a synopsis of the status of cyber-physical systems in manufacturing systems. In another study, Wang et al. [5] described what the system architecture and data-sharing infrastructure should be for controllability of cyber-physical systems on the cloud. Lee et al. [21] discussed an application of cyber-physical systems and smart algorithms in equipment maintenance. Cooper and Wachter [23] studied the application of cyber-physical systems for thermal modeling in the additive manufacturing processes. Seitz and Nyhuis [24] presented a learning factory idea and discussed the importance of cyber-physical systems for not only monitoring production, but also educating and training the employees in manufacturing. While the overall influence of cyber-physical systems as technological changeovers already has been discussed in the literature, current studies are still in the conceptual level stage [16].

Wang, Gao, and Ragai [5] presented an integrated cyber-physical system for cloud manufacturing. It introduced an architecture of cyber physical system for cloud manufacturing with three tiers: control tier, view tier with a web interface, and model tier. Three case studies are provided: the first one is about remote monitoring and control in a mini robotic cell with one serial robot, the second one is a Web-based distributed process planning, and the third one is 3D model-guided remote assembly. Its architecture is clearly for a cyber-physical manufacturing system with a web interface. But it is not a cyber physical manufacturing cloud since it does not provide a common virtualization method of various machining tools, which is essential in a cyber physical manufacturing cloud, and it is not based on a service oriented architecture. In addition, it does not use the RESTful communication protocols and standard web data exchange method, such as XML and JSON, for exchanging manufacturing data over the Internet. Consequently, it is not scalable for cyber physical manufacturing cloud over the Internet. Helu, and Hedberg presented an architecture of integrated cyber-physical systems for enabling smart manufacturing across the product lifecycle [6]. Its test bed consists of a Computer-Aided Technologies Lab and a Manufacturing Lab using product models as a basis of information exchange in product lifecycle. It is proposed that MTConnect be used for monitoring a production process. It uses web services to disseminate data and information in testbed. It is a cyber physical system for manufacturing covering many phases in a product life cycle. However, it is not cyber physical manufacturing cloud since it does not provide a common method of virtualization of manufacturing resources and it does not allow direct operation of machine tools over the Internet event though they can be monitored using MTConnect. Several cloud-based commercial manufacturing systems exist. Authentise [25] developed 3DIAX, a cloud based marketplace for 3D modelling services for the manufacturers. Authentise also developed a Manufacturing Execution System to provide human-in-the loop cloud manufacturing services. It allows customers to order 3D printing services. Once orders are received, human operators operate 3D printers for their production. Customers can monitor their production process over the Internet. However, 3D printers cannot be operated over the Internet from the cloud in these systems. As a result, they are not cyber physical manufacturing clouds.

Virtualization of manufacturing resources is important when it comes to CPMC. Liu et al [26] proposed an approach of virtualizing manufacturing resources by representing machine characteristics and service encapsulation using XML [27, 28, 29, 30, 5]. Liu, Yan, and Lin proposed a multilevel framework of manufacturing resource virtualization [31] with layers of representation of functional capabilities of machines based on a web ontology language. Zhao et al [32] represented machine characteristics and functionalities using RESTful web services. Ameri and Dutta [33] proposed a manufacturing service description language (MSDL). One of the major issues with the above virtualization approaches is that they consider either machine characteristics or machine capabilities, but not both.

Communication methods play an important role in CPMC. In general, RESTful web services are invoked by XML data formats. MTConnect is an XML based communication method for machining tools and it can be used for monitoring their status. Thereby, the use of XML data specified by the MTConnect standard has a potential for virtualization. Xu [2] proposed the use of MTConnect for cloud manufacturing. Vijayaraghavan and Sobel [34] explained the integration of manufacturing system using MTConnect detailing the machine specific data. Vijayaraghavan and Dornfeld [35] described the use of MTConnect for enabling process planning and verification in an industrial environment. MTConnect as a communication standard is viable for real-time data operations as shown by research of Michaloski and Venkatesh et el [36]. An alternative of MTConnect standard, TMTC (Taiwanese Machine Tool Connect), has been proposed by Lin et el [37]. The TMTC is a protocol where an integration of different types of Adapters was implemented in a single machine, increasing the scalability of the implemented system. Although MTConnect supports only machine monitoring operations, it cannot be directly used for operating machining tools in CPMC. MTConnect is being widely accepted commercially as a communication method of machine tools over the Internet. System Insights with ROS-Industrial developed a peer to peer inter machine communication architecture using MTConnect [38, 39]. The communication architecture is used for the communication between robotic arms and Computer Numerical Machines (CNC). Another peer-to-peer communication was established by LNS America in collaboration with Mazak Corporation using MTConnect [40, 41]. Lately, STEP Tools Incorporated with UI-LABS started a project of O3 to develop a web environment to enable the orchestration of machining and measurement processes from tables and smart phones [42, 43]. The

O3 project is still seeking for a solution by combining the data of STEP [43], Quality Information Framework (QIF) [44], and MTConnect standards into a unified stream that can be used to evaluate the overall quality of a product. Some other communication mechanisms were also used in several other projects besides MTConnect, such as – Transmission Control Protocol (TCP). Wang, Gao, and Ragai [18] used TCP as a primary communication protocol to monitor and control machine tools. ROY-G-BIV developed a machine communication platform called XMC for controlling machine tools over a network [45]. In our CPMC, the communication mechanism from cloud to the manufacturing machines are based on both MTConnect and TCP combined. Work on this chapter is a continuation of previous a series of researches on cyber-physical manufacturing clouds [46, 47].

1.3 Scalable and service-oriented layered architecture of CPMC

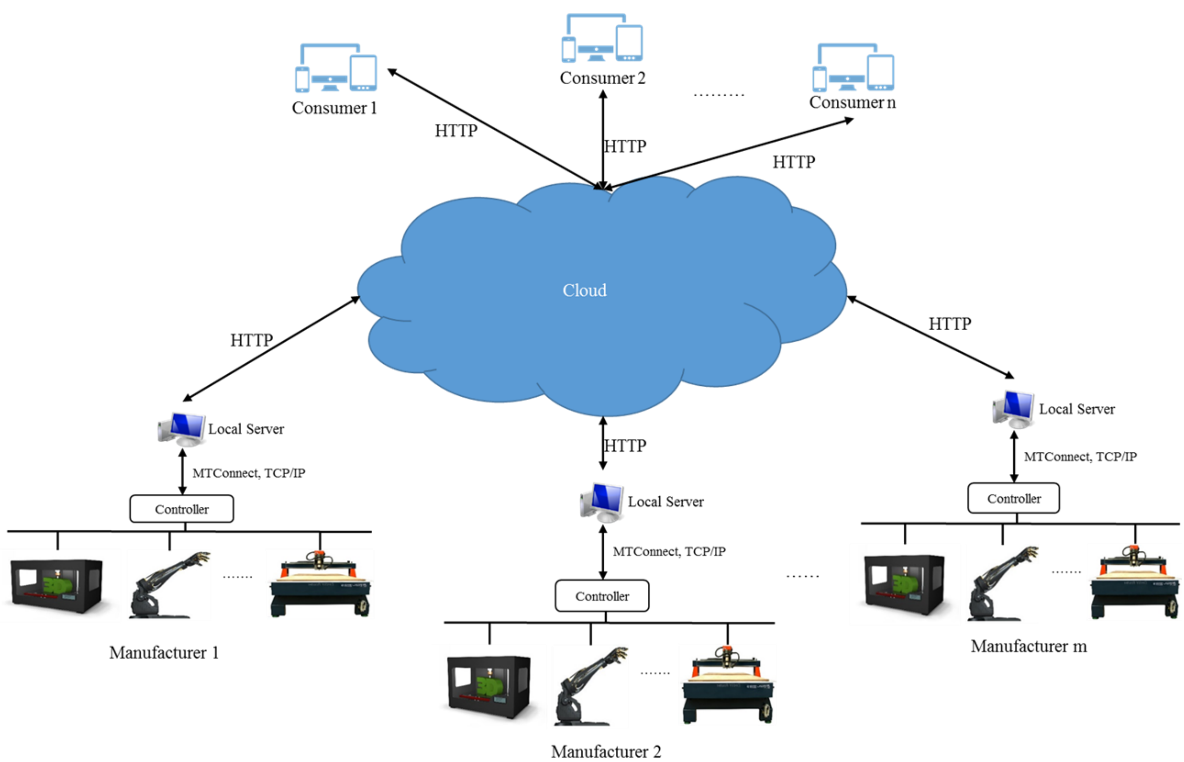


Figure 1: Conceptual model of Cyber-Physical Manufacturing Cloud (CPMC)

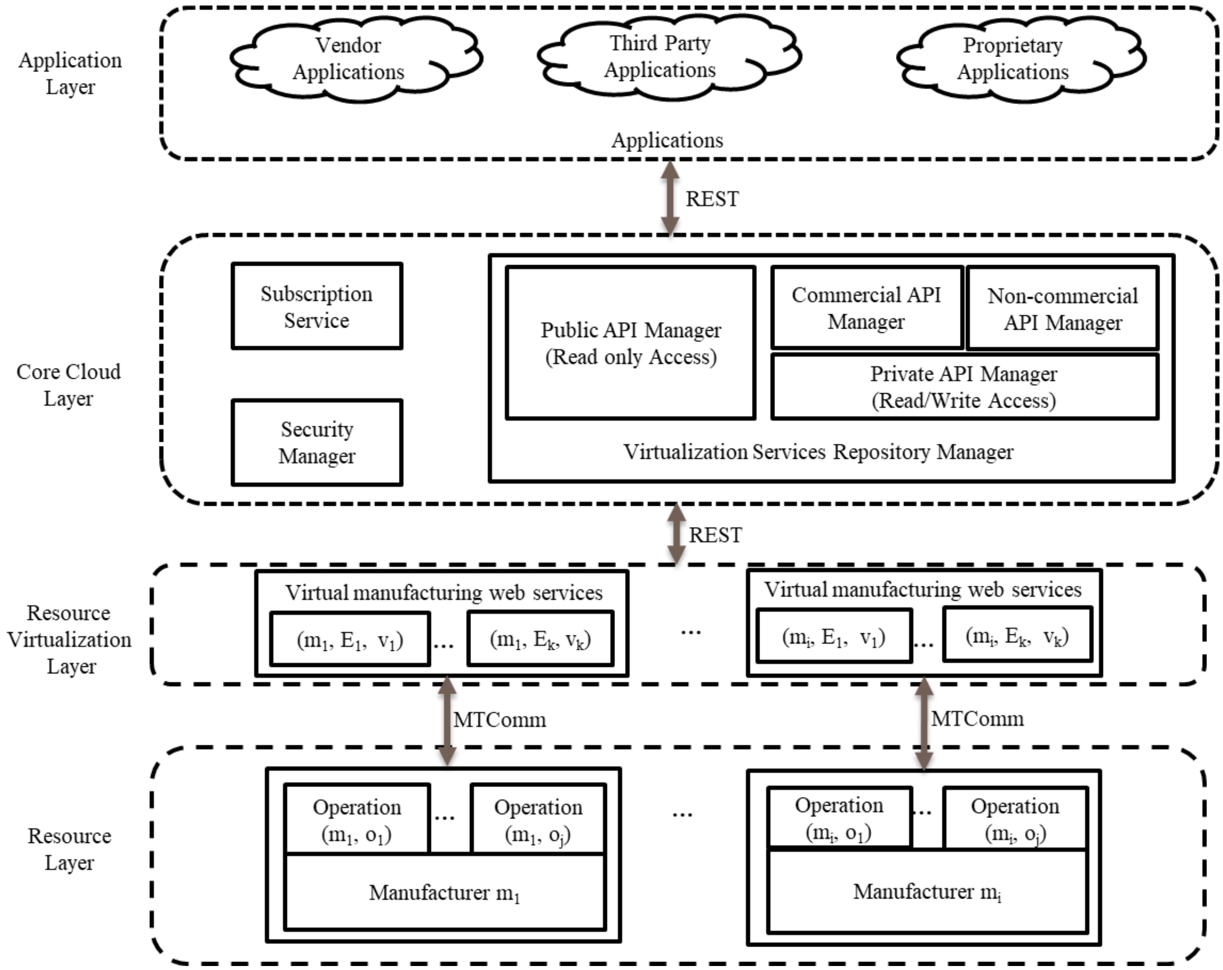


Figure 2: A four-layer architecture of CPMC.

A conceptual model of CPMC is presented in Figure 1. Customers communicate with the CPMC using multiplatform applications from desktops or mobile devices. The multiplatform applications use the secured Hypertext Transfer Protocol (HTTP) based communication protocol. Cloud manufacturing services are hosted in the cloud servers. The cloud services communicate with the manufacturers via local servers to transfer operational commands for monitoring machine tools and operating them. The manufacturers develop and host local servers to connect with the manufacturing tools on the factory floors using communication protocols such as TCP/IP (Transfer Control Protocol/Internet Protocol), RESTful MTConnect. Each manufacturing unit is considered as a CPS. Controllers are used to operate machining tools in each CPS in a local area network. The local servers forward the commands received from the cloud to the controllers for performing manufacturing operations.

1.3.1 Scalable service-oriented architecture of CPMC

Error! Reference source not found. presents a scalable service-oriented layered architecture of CPMC. The architecture presents a hierarchical view of cloud and communication services. In the diagram, the dotted rounded rectangles are the layers and the solid rectangles inside the layers are the components of a respective layer. The communications between the layers are shown in gray thick doubly pointed arrows. This architecture is discussed from bottom up.

- **Resource Virtualization Layer:** Resource virtualization layer is used to virtualizing manufacturing resources and represent them in CPMC. CPMC virtualizes manufacturing resources by publishing RESTful web services to be accessible over the Internet. In **Error! Reference source not found.**, virtualized manufacturing web services are a set of RESTful web services that can access over the Internet. The set of RESTful web services for manufacturing operations are developed and deployed by the manufacturers in the servers of the resource virtualization layer. A local server in the conceptual model of CPMC, as shown in Figure 1, serves as a virtualization server for a manufacturer. There virtualization servers host and run their manufacturing web services. Beside virtualization, resource virtualization layer secures the manufacturing resources by authenticating access requests over the Internet and validating incoming operation requests from the cloud. The resource virtualization layer communicates with the core cloud layer using HTTP based REST protocol.
- **Core Cloud Layer:** The core cloud layer hosts a set of basic cloud services such as user subscription manager, security manager, and manufacturing operation virtualization services manager. These cloud services are developed as RESTful web services using HTTP based

REST protocol for inter-component communications. The core cloud layer also communicates with Application Layer using HTTP based REST protocol. The components of the layer are described as follows.

- Subscription Service: The CPMC users are managed by this component. Users request for an authentication token from this component to operate machining tools over the Internet. Besides, this component has a database of the user's preferences and subscriptions.
- Security Manager: An authentication token is provided to CPMC users by the subscription service component and it is created, monitored, and evaluated by the security manager. The authentication tokens provided by this component are time-based and auto expiring in nature; thus, enhancing system security of the system.
- Virtualization Service Repository Manager: Virtualization service repository manager organizes and stores the applications and virtualization web services data. Each published virtualized RESTful web service of manufacturing resources has a Uniform Resource Locator (URL). URLs of the web services are stored with necessary parameter information in a database. Manufacturers can configure access of their published web services. These published web services are managed based on their access authorizations as follows.
- Public API Manager: The public API manager provides the information of the published web services by the manufacturers to execute the service from applications. The published web services are open for anyone in the web to use. Due to the open nature of this type of services, it is highly likely that they are used for monitoring.

- Private API Manager: Private API manager manages the use of the published web services by imposing a limited access as specified by the publishing manufacturers. The published services are classified in two categories:
- Commercial API manager: This type of web services is published for commercial use by the application developers based on subscription.
- Non-commercial API manager: This type of web services is published for non-commercial use by the application developers who are given access by the manufacturers to develop applications for internal maintenance purpose of the manufacturing environments.

- **Application Layer:** The application layer manages multiplatform applications for customers to perform manufacturing operations over the Internet. The platforms of the applications can be web browser, embedded systems, desktop operating systems, and mobile operating systems. Positioned at the top of the layered architecture, the applications use cloud services of the core cloud layer to perform manufacturing operations over the Internet. There are three categories of applications published in the application layer, as described below.

  i. Vendor applications: Vendor applications are generally developed by manufacturers for the public access of consumers of a CPMC.

  ii. Proprietary applications: Proprietary applications are developed by manufacturers for a limited set of consumers, such as paid users.

  iii. Third-party applications: Third-party applications are developed by organizations who subscribe and utilize the application programming interfaces (API) provided in a CPMC by the manufacturers.

1.4 Virtualization of manufacturing resources using RESTful web services

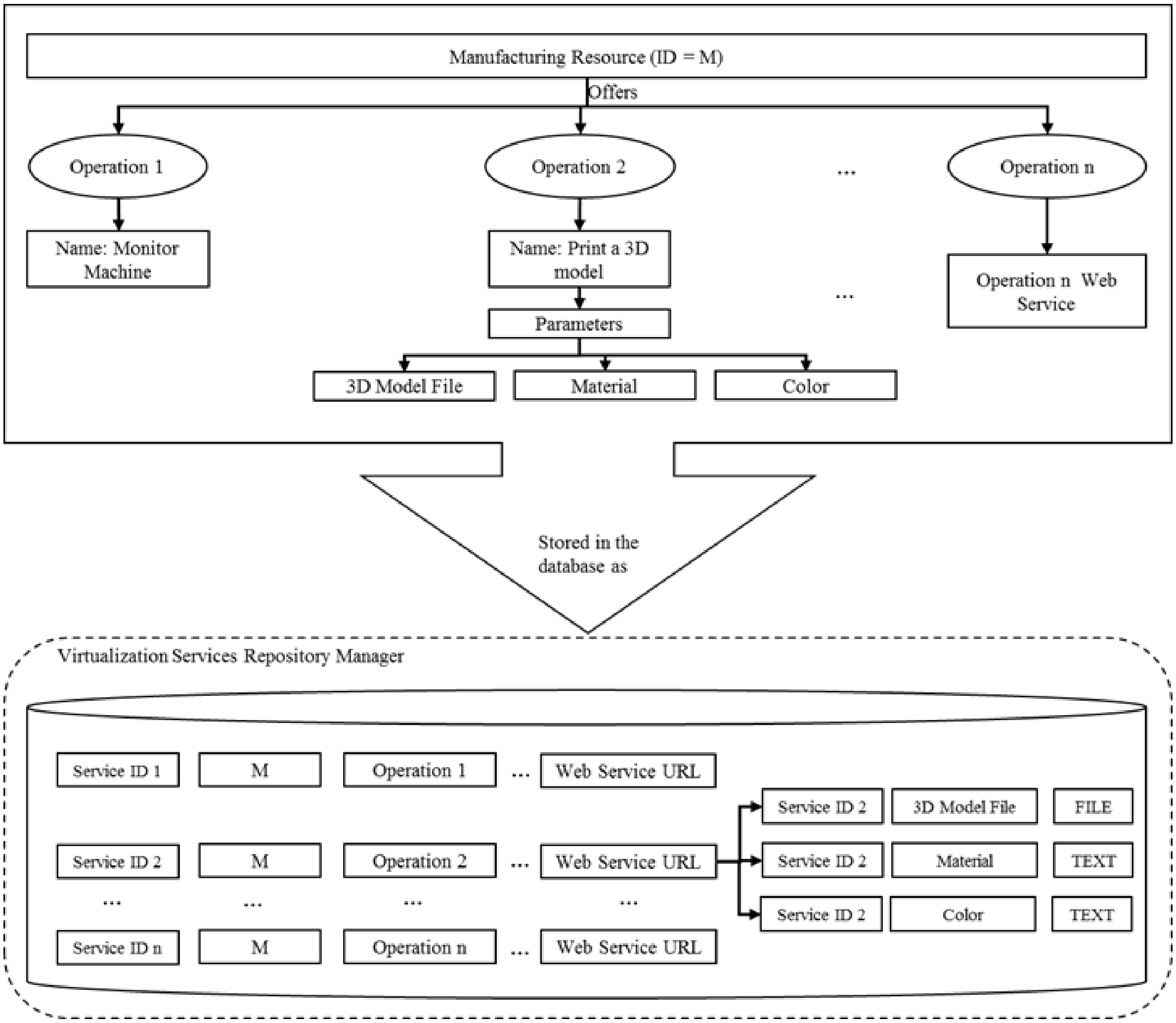


Figure 3: Workflow of Virtualization method in CPMC

Virtualization of manufacturing resources in the CPMC is done by developing and publishing manufacturing web services. These web services provide functionalities of the manufacturing resources. Each manufacturing web service is available through a Uniform Resource Locator (URL) accessible in the cloud. For example, a web service can be developed and made available for a 3D printer to display printing progress over the Internet. The virtualization method provided in this layered architecture of CPMC stores URLs of the web services with other Meta data about the manufacturing resources and functionalities. In our CPMC, a manufacturing resource has an ID. Each manufacturing resource can perform one or more operations through web service over the Internet. Figure 3 shows a workflow of the virtualization. A manufacturing resource M has n number of operations. Each operation is transformed into web services and each web service has an URL. A database of the web services is developed to manage virtualized manufacturing resources. A database server storing the web services is hosted in the cloud and is called Service Repository. In the layered architecture, Virtual Service Repository Manager (VSRM) component inside the Core Cloud Layer manages the service publishing. Manufacturing web services in Service Repository may have parameters and they are stored in the database. Security of the web services are maintained through an auto expiring encrypted security token-based mechanism. To access published web services stored in Service Repository, cloud-based applications must request for an authorization to the VSRM. The VSRM then provides the URL of the requested web services with the information of the parameters in JSON data format.


```
{
   "service_id":"1054265",
   "service_url":"http://local-server-ip:port/virtualization/ultimaker/control/print",
   "primary_category":"3D Printing",
   "secondary_categories":"Prototyping",
   "provider":"University of Arkansas",
   "ssl_support":"YES",
   "authentication_model":"API KEY",
   "params":[
      {
         "param_name":"cad_file",
         "param_value_type":"FILE"
```
(Continued from left)
```
      },
      {
         "param_name":"material",
         "param_value_type":"TEXT"
      },
      {
         "param_name":"color",
         "param_value_type":"TEXT"
      }
   ]
}
```


Figure 4: Virtualization data of a manufacturing resource using web service for 3D printing in JSON

Figure 4 presents a simple example of virtualization of a manufacturing resource using a web service for 3D printing in JSON format. Each web service in Service Repository is stored with an ID, URL, some other metadata, and security configurations. The example in Figure 4 shows how the parameters of the web service is managed. Each parameter has a name, type of data, and value. In this example, this web service needs a 3D model file to start a 3D printing operation. It has a parameter for the 3D object file and its type is FILE. When an application requests for this operation, the application must provide a CAD file as a 3D model, name of the material, and color of the object.

1.5 Publication and Subscription of Manufacturing Applications and Web Services

The service-oriented architecture of CPMC enables an integration of cloud-based applications with services in manufacturing environments. Additionally, the architecture provides a platform to develop Internet scale manufacturing application software and web services and make them available over the Internet.

1.5.1 Application and web service publishing center of the CPMC

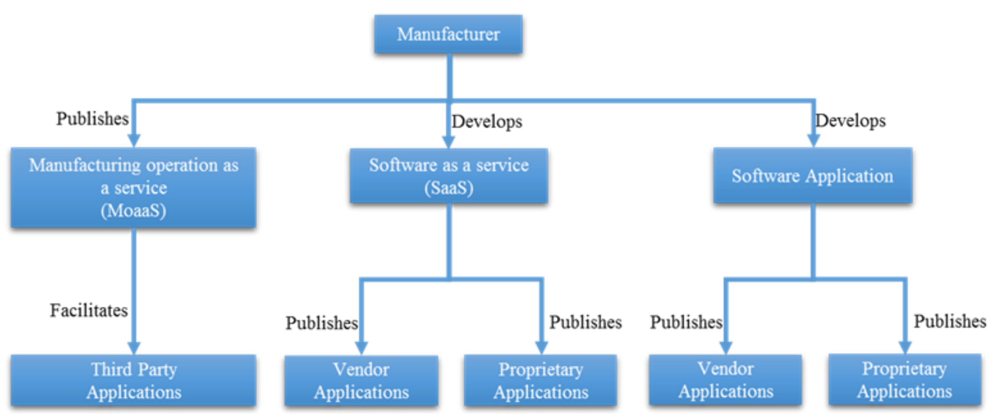


Figure 5: Application and services for manufacturing operations development hierarchy

Web service and application publishing center provides manufacturing operations as either web services or standalone software applications. The published manufacturing web services are accessible over the Internet based on their access privilege. Three kinds of access privileges are provided: public, commercial, and non-commercial web services. Figure 5 shows a classification of manufacturing services and applications over the Internet. An application and service publishing center is designed and developed in the testbed manage these types of services.

1.5.2 Workflow: Manufacturers, Application Developers, and Consumers

The roles of the participants in the CPMC include manufacturers, application developers, and consumers. Figure 6 shows a workflow from perspectives of these three participants. In the top portion of Figure 6, manufacturers develop manufacturing web services. The next step is that the manufacturers publish these services in the cloud. Once the services are available over the Internet, they are used to develop manufacturing applications over multiple platforms. Upon applications are developed, they may or may not be published in the application center. A workflow for CPMC consumers is described in the middle portion of Figure 7. In the bottom portion of Figure 7, third party software developers can develop manufacturing applications by subscribing the published manufacturing web services and using the web services in their applications.

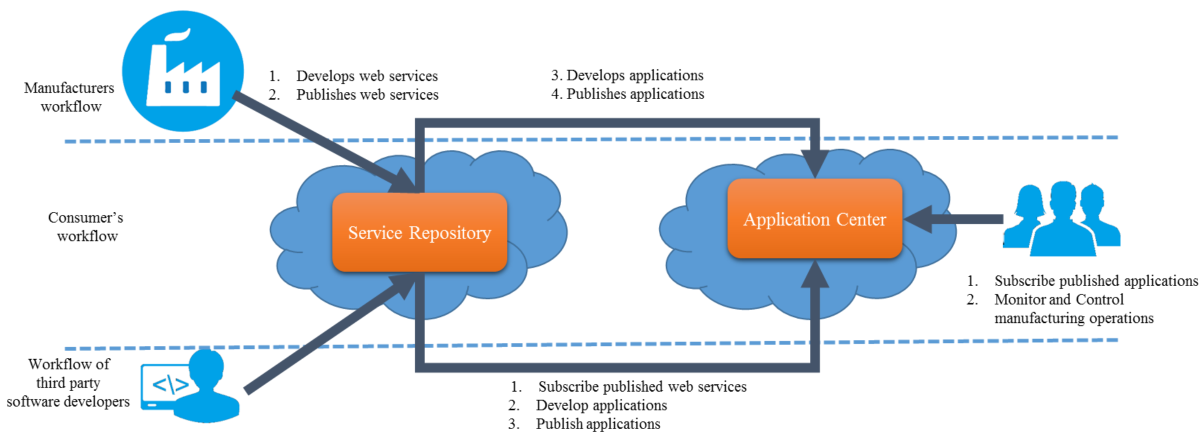


Figure 6: Workflow of the participants in the CPMC system

1.6 Testbed implementation

We developed a testbed based on the architecture of CPMC in Section 1.3. The testbed is developed as an operational prototype.

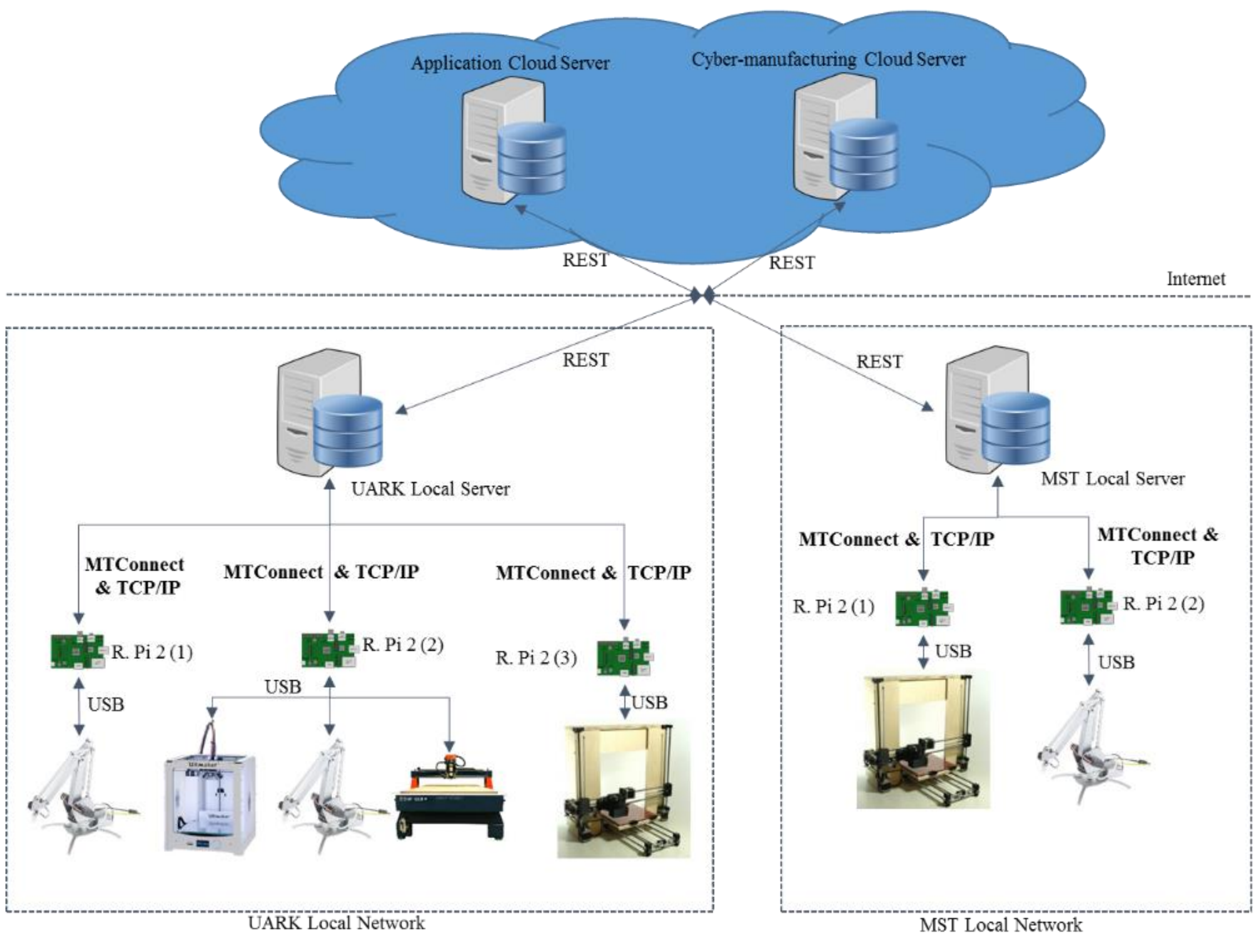


Figure 7: Testbed implementation of CPMC

1.6.1 Infrastructure implementations

We developed a testbed based on the presented architecture of CPMC. The testbed is developed as a full-scale operational prototype of the proposed model of CPMC. A structure of the testbed in presented in Figure 7. It has two manufacturing sites connected to the cloud over the Internet. Each site has a local area network of machining tools. The first one is in the University of Arkansas and the other is in the Missouri Institute of Science and Technology. The site in the University of Arkansas has five manufacturing machines, including one CNC, two 3-D printers, and two robotic arms, and three controllers, and one local server. The site in the Missouri University of Science and Technology has one 3-D printer, one robotic arm, two controllers, and one local server. Machining tools are connected to a controller and all the controllers are connected to the local server where the virtualized manufacturing web services are hosted. The local server is a communication focal point from the cloud. The communication from the local server to the controllers are implemented using MTConnect and TCP/IP protocol. The testbed uses MTConnect for monitoring manufacturing operations and TCP/IP for communicating manufacturing operations.

As explained in the conceptual model, the local servers host manufacturing web services. They receive and process incoming requests for manufacturing operations from the cloud. There are two cloud servers for hosting cloud-based applications and manufacturing cloud services respectively. The server for the manufacturing cloud services hosts the components of core cloud layer in the layered architecture. The application cloud server hosts an application center and several other cloud-based applications. The application center, which is a web application hosted in the cloud, is a marketplace of all the published applications. Users can sign into the application center and subscribe cloud-based applications. A screenshot of the application subscription page is shown in Figure 8. In Figure 9, a screenshot of the application center dashboard is presented. The dashboard contains access links of the subscribed applications of users. A mobile application developed for both iOS and Android operating system were developed as an example of the open framework for the third-party application developers. This vision behind the open framework of applications is to allow anyone to develop cloud-based applications for manufacturing and they can be hosted in the cloud. A screenshot of a smartphone manufacturing application developed for the testbed of CPMC is presented in Figure 10. The mobile application can be used to monitor a manufacturing tool and perform manufacturing operations over the Internet. To facilitate publishing web services by the manufacturers, a web-based application named Web Service-

Publishing Center is developed. Manufacturers can publish their web services as illustrated in Figure 11. The published web services then appear on the list of services as shown in Figure 12 in the Service-Publishing Center search web page. Users can search for web services by functionalities and other search criteria.

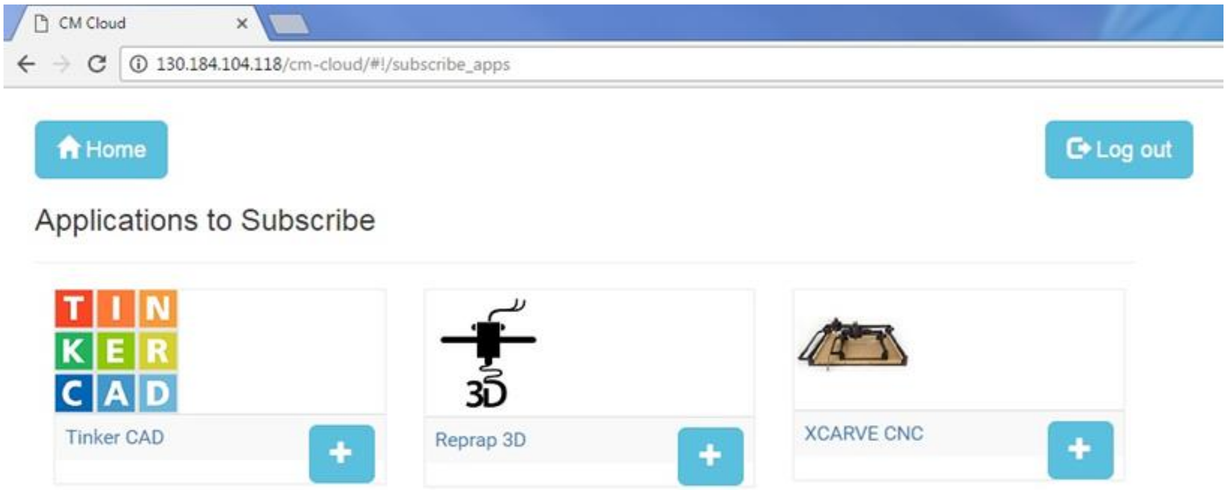


Figure 8: Subscribe applications in the Application Center

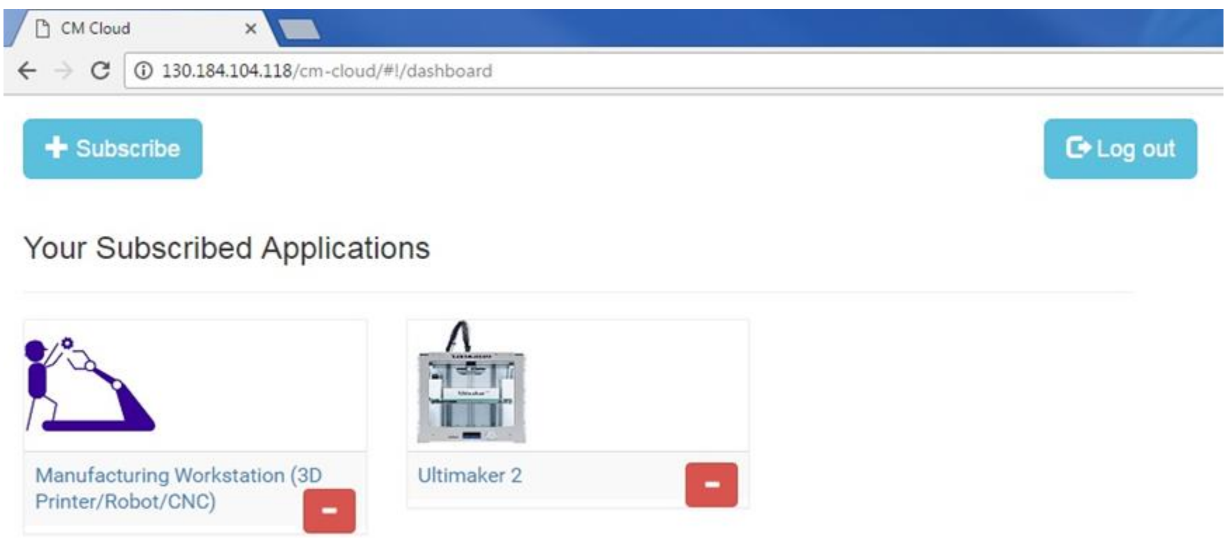


Figure 9: Application Center dashboard showing subscribed applications

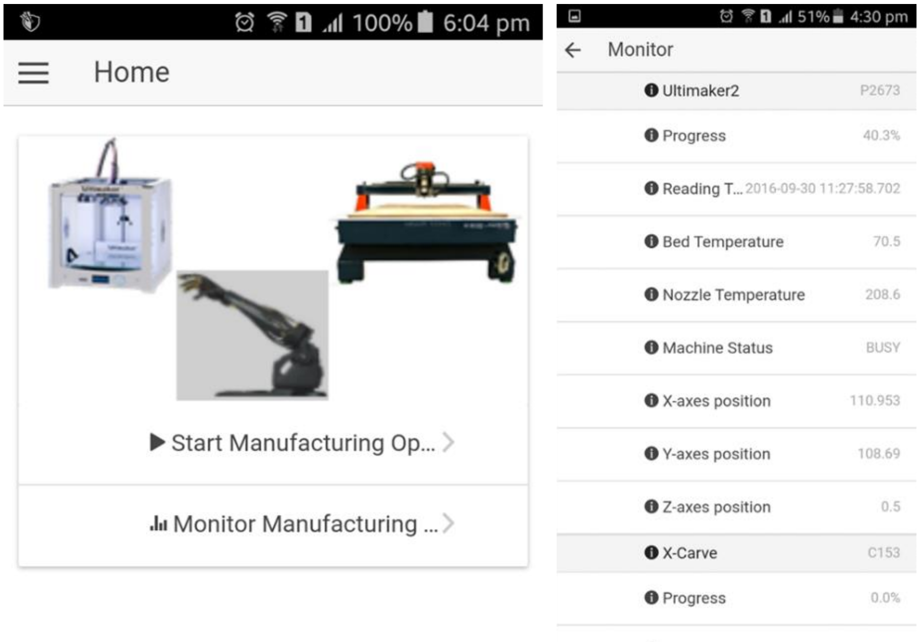


Figure 10: Dashboard and Monitoring Operation view of a third-party smartphone application

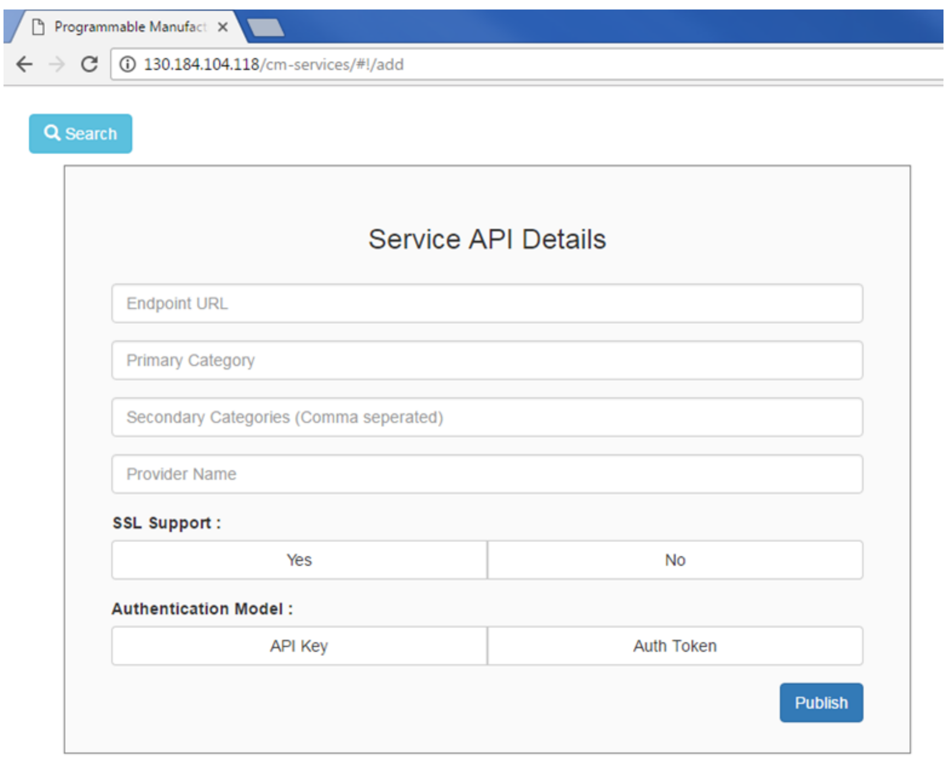
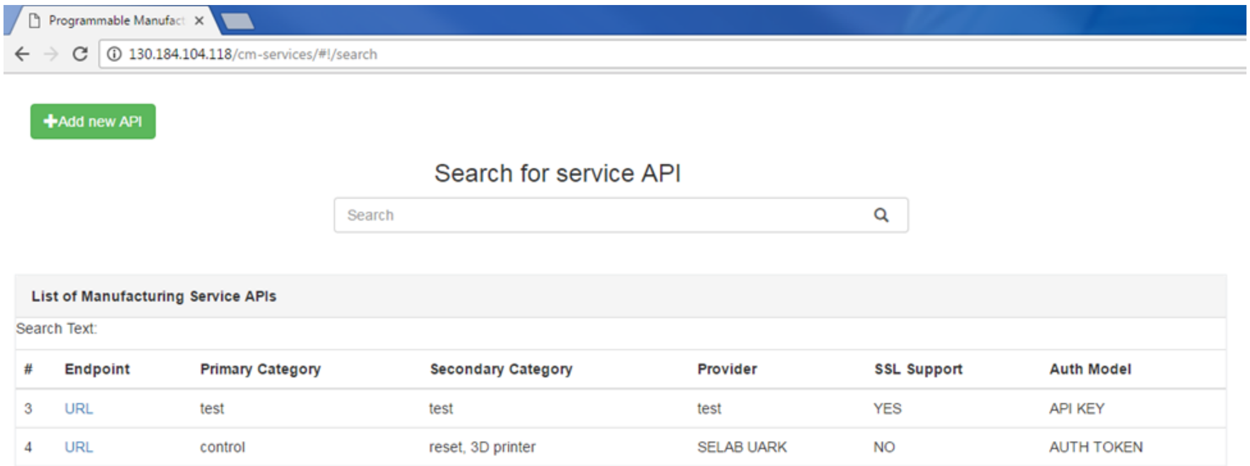


Figure 11: Service-Publishing Center – Publish a web service



#	Endpoint	Primary Category	Secondary Category	Provider	SSL Support	Auth Model
3	URL	test	test	test	YES	API KEY
4	URL	control	reset, 3D printer	SELAB UARK	NO	AUTH TOKEN

Figure 12: Search Engine of published web services

1.6.2 Evaluation of the testbed in manufacturing application scenarios

The testbed is evaluated using many manufacturing application scenarios. Two of the evaluated manufacturing scenarios are presented here to demonstrate capability and performance of our CPMC testbed.

1.6.2.1 Manufacturing application scenario 1: Directly operate a 3-D printer from cloud over the Internet

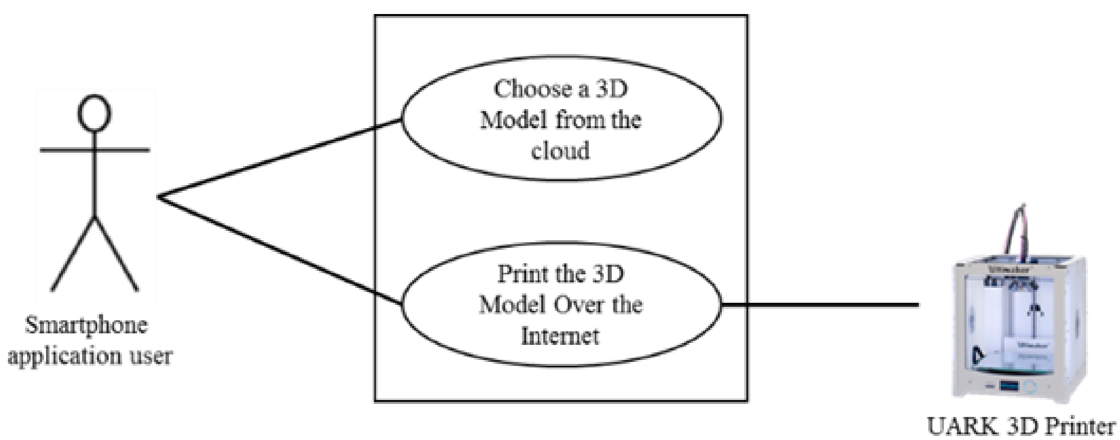


Figure 13: Use Case diagram for manufacturing application scenario 1

In this manufacturing scenario, a smartphone manufacturing application implemented in the testbed is used to perform a manufacturing operation using a 3D printer from cloud over the Internet. This manufacturing application scenario described as a Use Case in *Figure 13,* shows a customer can use smartphone to operate a 3-D printer from our CPMC over the Internet. **Error! Reference source not found.** contains a sequence diagram to show how components in our CPMC work together to implement this manufacturing operation over the Internet. The 3D printer is actively operated in the University of Arkansas (UARK) from our cloud. Clearly, this operation requires streaming a 3D CAD model file to the UARK 3D printer. The smartphone application in the CPMC sends the file along the print operation request to the local server. The local server validates the request and forwards it to the controller of the machine. The controller then parses the request and instructs the machine to perform the operation accordingly.

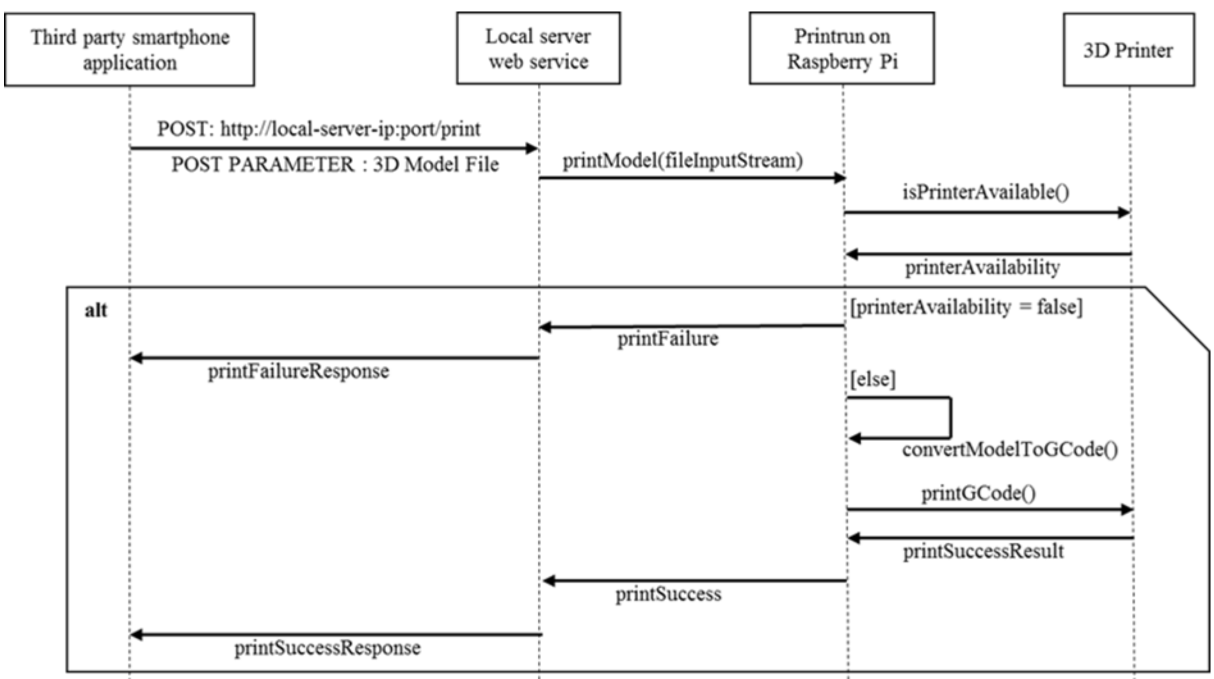


Figure 14: Sequence Diagram for manufacturing application scenario 1

1.6.2.2 Manufacturing application scenario 2: Monitoring operations of 3D printer from the cloud over the Internet

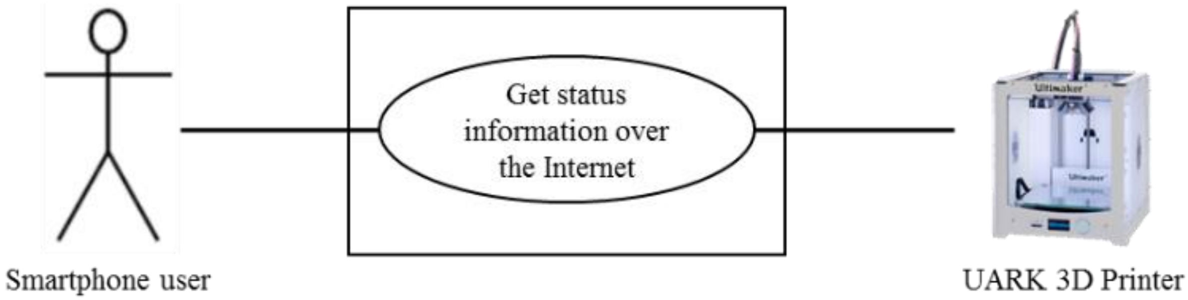


Figure 15: Use Case diagram of manufacturing application scenario 2

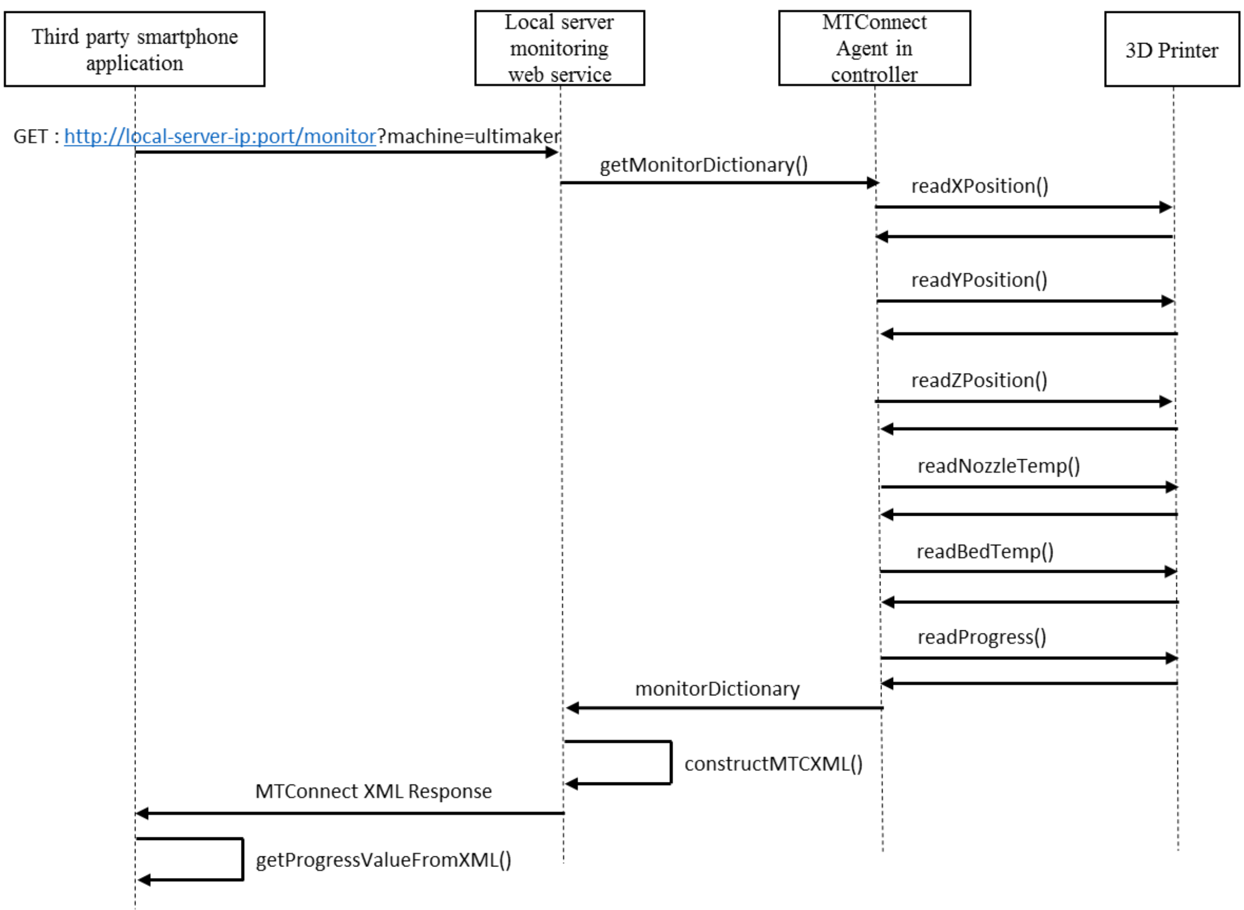


Figure 16: Sequence Diagram for manufacturing application scenario 2

When the printing job is in progress, a user can use our CPMC to monitor the printing progress of the 3-D printer. This manufacturing application scenario is represented as a Use Case in Figure 15. A sequence diagram in Figure 16 shows how components in our CPMC testbed work together to monitor progress of a printing job performed by a 3-D printer on a real-time basis.

The user sends a request for monitoring the 3D printer using a smartphone manufacturing application implemented in our CPMC testbed. Upon authorization from the Subscription service of the CPMC, the smartphone manufacturing application requests the local server to provide status information using MTConnect protocol. Upon a validated incoming request from the local server, an MTConnect agent gathers data from the 3D printer through an MTConnect adapter and returns it to the local server in XML format using MTConnect for further processing. The local server then transmits the data to JSON format and returns to the smartphone application to display to the user.

1.6.3 Result Discussion

The proposed architecture of CPMC provides a strong support to the scalability of manufacturing. Manufacturing resources can be virtualized and added to the CPMC. They even can be reconfigured dynamically. A manufacturer can develop web services to enable accessing manufacturing services over the Internet and publish them in the cloud. To remove manufacturing resources, all a manufacturer needs to do is to remove their web services from the web service publish center in our CPMC. Although the CPMC architecture and testbed show feasibility of integration of manufacturing cloud and cyber physical systems to enable directly operation and monitoring of machining tools from a cyber physical manufacturing cloud over the Internet, we still need to perform additional researches and experiments on large-scale cyber physical manufacturing clouds before achieving its potential in the industry. The layered service-oriented architecture for CPMC is the first step towards an open-architecture of cloud-based manufacturing systems. Many challenging issues need to be investigated in this direction. Few of the issues are -

security of CPMC based on an open architecture, production process planning, dynamic configurability of the manufacturing resources, and big data analytics of manufacturing.

This research on CPMC provides valuable insights on the development of an Internet-scale manufacturing communication protocol. MTConnect standard and TCP/IP protocol are used to connect manufacturing tools over the Internet. However, by design, MTConnect cannot be used for direct communication of manufacturing services over the Internet due to its focus on data acquisition for monitoring from machining tools. Instead, TCP/IP is utilized as a workaround for communication of manufacturing services in the testbed. However, it does not satisfy several requirements of Internet scale manufacturing communication methods for easy and efficient exchange of manufacturing services over the Internet. More researches need to be done on Internet scale manufacturing communication method.

Speedy transmission of manufacturing data exchange across layers is crucial for a successful implementation of a CPMC. Not all existing industrial machine tools are capable of computation and networking. Besides, many existing machine tools are operated by outdated operating systems. Although addition of those machines to a CPMC is challenging, we had a preliminary success with several types of those in the development of the testbed of CPMC. In the testbed of CPMC, several machine tools, such as one CNC, two 3D printers, two robotic arms did not have the networking capabilities except the USB interface. We added low-cost Raspberry Pi 2 controllers with MTConnect to connect them through USB interfaces to the network. Respective driver programs are also installed in the controllers to monitor and operate these machines. However, for large industrial machine tools, small Raspberry Pi 2 controllers may not be enough. Therefore, more researches need to be conducted to develop networking and middleware technologies that enable machine tools with outdated operating systems and hardware to be networked and their services can be provided in a CPMC.

1.6.3.1 Machine monitoring performance

In our implementation, the MTConnect Agent that runs in a Raspberry pi 2 machine, is a tiny RESTful server and it collects machine status information from the manufacturing resources. The response time of the Agent over the Internet was tested for performance evaluation. We tested the response time of the Agent web services using Apache Benchmark which is a renowned test framework for web services response time. The results are mentioned in the table of Figure 17. The performance analysis shows that the web services offered by the Agent has ~5.5 milliseconds of average response time, which proves the real-time capability of machine status monitoring operations presented in the architecture.

Service name	Time taken for tests (seconds)	Complete requests	Failed requests	Total transferred (bytes)	Requests per second (mean)	Time per request (mean in ms)
Probe	5.452	1000	0	115000	183.43	5.452
Current	5.714	1000	0	115000	175.00	5.714
Sample	5.379	1000	0	115000	185.92	5.379

Figure 17: Response time of MTConnect services for monitoring operations

1.6.3.2 Machine operation performance

In the implementation, the operational commands are sent as data bytes in a TCP/IP networked communication. Although the communication for control commands is in a preliminary stage, bytes communication for commands and file streaming for CAD models are done efficiently.

From an experiment, a sample 3.89 MB sized CAD file sent to the Ultimaker 3D printer across University of Arkansas local network via the Raspberrry Pi 2 Controller in 1585.33 milliseconds resulting in a file transfer rate of 2.45 megabytes per seconds (MBps). Rest of the control commands are in generally 4 to 10-character bytes in size that takes very small amount of time. The total time to complete the operations depend on the manufacturing machines.

1.6.4 State-of-the-art comparison

Cyber-Physical Manufacturing Cloud (CPMC) virtualizes monitoring and operations of physical manufacturing resources through web services. It establishes direct and two-way communication between virtualized machines in cloud and their physical counterparts in shop floors. In summary, CPMC includes the following features. Firstly, it provides a service-oriented architecture that supports high scalability of heterogeneous manufacturing resources. Secondly, it enables direct monitoring and operation of machine tools over the Internet. Thirdly, it creates virtual copies of heterogeneous manufacturing resources in manufacturing clouds by a uniform virtualization method. Fourth and finally, it diagnoses system health and test machine tools at a granular level over the Internet in order to diagnose and detect faults [37]. Features of CPMC can be contrasted with another popular manufacturing concept called cyber physical production systems (CPPS) [10]. Both CPMC and CPPS integrate cyber physical systems (CPS) with machine tools. However, CPMC is different than CPPS because it integrates Cloud Computing and CPS with manufacturing resources. Unlike CPPS, CPMC virtualizes machine tools through web services in manufacturing clouds. In addition, all applications and core components of CPMC are hosted in clouds. In this chapter, we present a service-oriented DT architecture based on the features of CPMC.

1.7 Conclusion

Research of this chapter presents a scalable service-oriented architecture, virtualization technique, and communication methods of a cyber-physical manufacturing cloud. It integrates paradigms of cloud manufacturing and cyber physical systems to enable direct operation of machining tools from a manufacturing cloud over the Internet. It has a web-based application for publishing web services for manufacturing operations and an application center to publish software applications for manufacturing operations. It utilizes the RESTful Internet protocol MTConnect for acquiring status data of machining tools to monitor their operations over the Internet. It provides a TCP/IP based solution for communicating operational manufacturing services in the cyber physical manufacturing cloud. The implemented operational CPMC testbed demonstrates feasibility of monitoring machining operations and performing manufacturing operations and processes directly from a manufacturing cloud over the Internet.

CHAPTER 2: VIRTUALIZING PHYSICAL MACHINE OPERATIONS INTO THE DIGITAL TWINS IN CYBER-PHYSICAL MANUFACTURING CLOUD

Abstract **–** Digital Twins simulate physical world objects by creating as-is virtual images in a cyberspace. In order to create a well synchronized digital-twin simulator in manufacturing, information and activities of a physical machine need to be virtualized. Many existing digital twins stream read-only data of machine sensors and do not incorporate operations of manufacturing machines through Internet. In this work, a new method of virtualization is proposed to integrate machining data and operations into the digital twins using Internet scale machine tool communication method. A fully functional digital twin is implemented in CPMC testbed using MTComm and several manufacturing application scenarios are developed to evaluate the proposed method and system. Performance analysis shows that it is capable of providing data-driven visual monitoring of a manufacturing process and performing manufacturing operations through digital twins over the Internet. Results of the experiments also shows that the MTComm based digital twins have an excellent efficiency.

2.1 Introduction

Modern manufacturing systems are integrating cloud computing technologies deeply into the production processes. Consequently, virtualization of manufacturing resources in the cloud is finding a wide range of applications. Machine tool virtualization method facilitates creating virtual copies of physical machines in a cyberspace by mirroring real-time machine states and activities. Digital twins of physical machines, also known as virtual copies in some literatures, are created by the help of virtualization methods. They are formed by streaming machining data from the manufacturing sites. Some of the digital twins provide 3D visualization of physical machines remotely through internet. All the existing digital twins from our knowledge focusses only on read-only data from the physical machines. Consequently, existing digital twins cannot perform manufacturing operations from a remote environment. If we try to envision a scenario where a manufacturer owns a shop floor in a geographically distant location and wants to visually operate every manufacturing processes from digital twins of the shop floor machines remotely over the internet, it cannot be done with the existing digital-twin systems. In order to enable the vision of virtually operating a shop floor, we virtualize and integrate manufacturing operations into the digital twins of physical machines at a granular level.

To enable cloud-based digital twins operate physical machines, two key problems are identified:

a) virtualization of physical machine tool operations in the cloud
b) integration of the virtualized manufacturing operations with the digital twins.

Problem (a) is solved in this work by using MTComm (mtcomm) method as the manufacturing communication protocol. MTComm virtualizes manufacturing operations of a machine by transforming into RESTful web services. It also virtualizes status information of the machine operations. The developed digital twin of this work take advantage of MTComm based communication with physical machines and be able to operate [48, 49, 50]. However, only virtualizing manufacturing operations is not enough. A machine tool can have multiple functionalities and operations. Some operations are relevant to the whole machine while some are relevant to specific machine components. Problem (b) addresses this issue of integrating virtualized manufacturing operations into digital twins. To solve this problem, this work presents a method to integrate virtualized manufacturing operations at a granular level into the digital twins. The developed digital twin of this work constructs a 3D visual model from the MTComm based ontology information. The 3D visual digital twin is rendered in manufacturing applications having integrated each virtualized part of the machine with specific manufacturing operations and tasks.

Several cloud-based digital-twin models are proposed in a numerous number of researches. Despite that, there remains a gap between digital twins and physical machines. Here the gap is minimized by turning manufacturing resources into cyber-physical machines and bring closer to cloud. A new paradigm of cyber-physical manufacturing cloud (CPMC) is presented. CPMC virtualizes manufacturing resources in cloud. It establishes bi-directional and direct communication between virtual copies and physical machines in cloud environments. In a nutshell, CPMC includes the following features: (a) provides a service oriented architecture that supports high scalability of heterogeneous manufacturing resources; (b) enable direct monitoring and operation of machine tools over the Internet; (c) create virtual copies of heterogeneous manufacturing resources in manufacturing clouds by a uniform virtualization method; and (d) diagnose system health and test machine tools at a granular level over the Internet in order to detect faults.

A 3D visual digital twin is developed in CPMC to operate machines remotely over the internet. MTComm is deployed as the primary communication method of CPMC so that machine operations are virtualized in the cloud. In order to ensure granular level integration of the virtualized manufacturing services, MTComm method ontology is expanded to support providing 3D models of the machine components. The developed digital twin is rendered in multiplatform client applications. It can perform operations of machine level such as turn on/off a 3D printer machine or component level such as move Axis of the machine. Machine read-only data is also streamed to the digital twin in order to implement a fully functional prototype in the CPMC testbed. Performance of the implemented digital twin is evaluated by experimenting it in five manufacturing scenarios. 3D visual performance of the digital twin is analyzed by calculating network delays between machine data reading time and rendering time in client applications.

Network delay varied from 100ms-6seconds averaging at 0.38s. The following points are the primary research contribution of this work:

i. Developing an MTComm based virtualization method to enable operating machining tools from the digital twins.

ii. Integrating virtualized manufacturing operations with the 3D visual digital twins at a granular level.

2.2 Literature Survey

2.2.1 Service-oriented Virtualization and Communication methods of physical machine tools

Virtualization of manufacturing resources is important when it comes to the objectives of this research. Liu et al [51] proposed an approach of virtualizing manufacturing resources by representing machine characteristics and service encapsulation using XML [27, 28, 29, 31]. Liu, Yan, and Lin proposed a multilevel framework of manufacturing resource virtualization [52, 53] with layers of representation of functional capabilities of machines based on a web ontology language. Zhao et al [32] represented machine characteristics and functionalities using RESTful web services. Ameri and Dutta [33] proposed a manufacturing service description language (MSDL). One of the major issues with the above virtualization approaches is that they consider either machine characteristics or machine capabilities, but not both.

Communication methods play an important role in CPMC. In general, RESTful web services are invoked by XML data formats. MTConnect is an XML based communication method for machining tools and it can be used for monitoring their status. Thereby, the use of XML data specified by the MTConnect standard has a potential for virtualization. Xu [2] proposed the use of MTConnect for cloud manufacturing. Vijayaraghavan and Sobel [34] explained the integration of manufacturing system using MTConnect detailing the machine specific data. Vijayaraghavan and

Dornfeld [35] described the use of MTConnect for enabling process planning and verification in an industrial environment. MTConnect as a communication standard is viable for real-time data operations as shown by research of Michaloski and Venkatesh et el [36]. An alternative of MTConnect standard, TMTC (Taiwanese Machine Tool Connect), has been proposed by Lin et el [54]. The TMTC is a protocol where an integration of different types of Adapters was implemented in a single machine, increasing the scalability of the implemented system. Although MTConnect supports only machine monitoring operations, it cannot be directly used for operating machining tools in CPMC. MTConnect is being widely accepted commercially as a communication method of machine tools over the Internet. System Insights with ROS-Industrial developed a peer to peer inter machine communication architecture using MTConnect [38, 40, 39, 55]. The communication architecture is used for the communication between robotic arms and Computer Numerical Machines (CNC). Another peer-to-peer communication was established by LNS America in collaboration with Mazak Corporation using MTConnect [41]. Lately, STEP Tools Incorporated with UI-LABS started a project of O3 to develop a web environment to enable the orchestration of machining and measurement processes from tables and smart phones [42]. The O3 project is still seeking for a solution by combining the data of STEP [43], Quality Information Framework (QIF) [44], and MTConnect standards into a unified stream that can be used to evaluate the overall quality of a product. Some other communication mechanisms were also used in several other projects besides MTConnect, such as – Transmission Control Protocol (TCP). Wang, Gao, and Ragai [5] used TCP as a primary communication protocol to monitor and control machine tools. ROY-G-BIV developed a machine communication platform called XMC for controlling machine tools over a network [45]. In our CPMC, the communication mechanism from cloud to the manufacturing machines are based on both MTConnect and TCP combined.

2.2.2 Virtualization of physical machines through Digital Twins

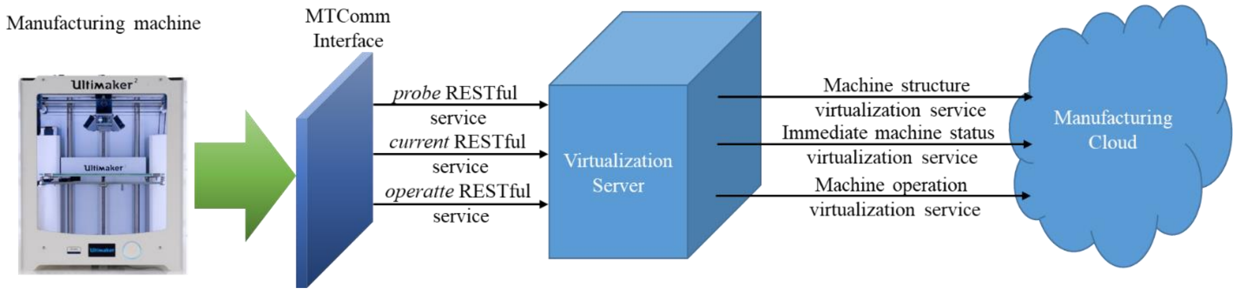
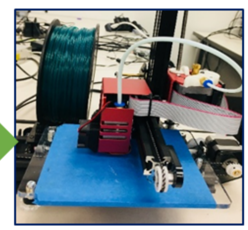


Figure 18: Method of virtualizing a manufacturing resource using MTComm [48]

The term digital twin was introduced publicly by Grieves nearly two decades ago [56]. In 2011, NASA defined DTs as an integrated multi-physics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin [8]. Tao et al. designed a DT driven product design, manufacturing and service with the help of big data [57]. Despite making significant research contributions, these literatures do not virtualize manufacturing operations. Kai et al. [58] presented a DT based cyber-physical production reference model for automated smart shop floors. In their testbed, they have integrated RFID sensors and computing devices for remote monitoring and operations. Their smart shop floor model dynamically allocates manufacturing subtasks to different machine tools or their specific parts in an optimized schedule. However, their system does not virtualize machine operations in cloud because they are only accessible inside a local cyber network. Their DTs lack proper data to achieve cloud-based manufacturing collaborations. Operable DTs of CPMC virtualizes machine operations in cloud and establishes collaborative manufacturing in cloud. Cloud manufacturing systems require virtual representation of physical manufacturing resources. Virtualization by service-oriented architecture is a common approach. Liu et al [31] proposed an approach of virtualizing manufacturing resources by representing machine characteristics and service encapsulation using XML. MTConnect [59] is used for virtualizing machine status information using its RESTful web services [60]. Liu et al. proposed a service repository-based method to virtualize machine operations [47]. One can utilize machine information from the MTComm method to virtualize physical machines in cloud. MTComm provides a web service, which describes characteristics and capabilities of machine tools in an XML document. Characteristics information include machine identity, type of machine, and description of real-time data items and so on. Capabilities information include machine operations, its parameters, and other production quality parameters. Existing virtualization models only virtualize machine characteristics. This method virtualizes capabilities and offers additional virtualization data for the operable DTs to collaborate with other DTs in cloud.

```xml
<Operation category="JOB" id="startJob" name="Start new job" type="PRINTING">
            <Parameters>
              <Parameter id="material" name="Material type" supportedMaterial="PLA"
type="MATERIAL" />
              <Parameter id="quantity" name="Number of objects" type="QUANTITY" />
              <Parameter id="objName" name="Object Name" type="OBJECT" />
            </Parameters>
          </Operation>
          <Operation category="ACTION" id="stopJob" name="Stop current job"
type="EXECUTION" />
          ...
            <Components>
              <Linear id="y" name="Y">
                <DataItems>...</DataItems>
                <Operations>
                  <Operation category="ACTION" coordinateSystem="MACHINE"
id="moveY" name="Move Y Axis" type="POSITION" units="MILLIMETER" />
                </Operations>
              </Linear>
...
```



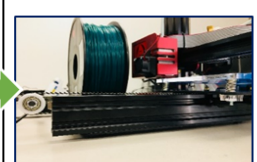


Figure 19: Mapping of manufacturing operations on machine level and machine

2.3 Method of virtualizing manufacturing operations using MTComm in CPMC

Both characteristics and capabilities information of a machine is collected from MTComm method. Figure 18 presents the method of virtualizing a manufacturing machine in CPMC using MTComm.

2.3.1 Virtualization of manufacturing operations using MTComm

MTComm [48] probe XML document not only provides data streaming information but also describes the operations can be conducted over the Internet. For developing an interactive virtual copy of a physical machine, manufacturing operations are also required to be incorporated in the virtualization. Figure 19 illustrates the virtualization of manufacturing operations on device level and device component level. Printing is a machine level operation that is mapped to the Bukito 3D printer machine. Similarly, a single component operation is mapped to the physical Linear Y axes component which moves the bed of the 3D printing machine.

2.3.2 Virtualization of manufacturing operation information of machining tools in CPMC


```
...
      <DataItems>
          <DataItem category="EVENT" id="availability" name="availablility"
type="AVAILABILITY" />
      </DataItems>
      <Operations>… <Operations/>
      <Components>
        <Axes id="axes" name="Axes">
          <Components>
            <Linear id="x" name="X">
              <DataItems>
...
                  <DataItem category="SAMPLE" coordinateSystem="MACHINE"
id="xPos" name="Actual X Position" subType="ACTUAL" type="POSITION"
/>
...
```

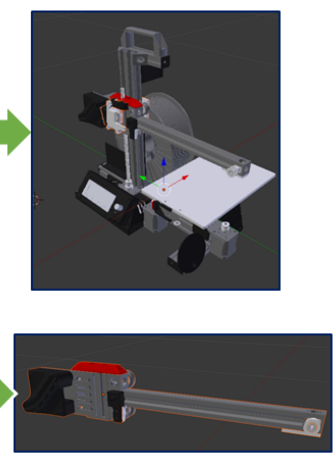



Figure 20: Mapping of real-time manufacturing operation status data streaming

To develop a virtual copy in CPMC, status information of machining tools needs to be synchronously reflected in near real time. MTComm can provide real-time machine status data which is utilized to synchronize the physical and virtual copy of manufacturing resources. In the probe XML document, data items which can be streamed into the virtual model is specified with necessary description. The job of the Virtualization Service in CPMC is to extract information set of those data items from the probe XML document and map accordingly to the virtual copies of the machines. Figure 20 explains the process of virtualizing real-time status of machine tools. Availability information of the machine maps to the machine level operation information. X axes position maps to the Linear X Axes component simulation.

2.4 Integration of virtualized manufacturing operations into 3D visual digital twin in CPMC

Integrating virtualized manufacturing operations with 3D visual digital twins requires two steps. First, 3D models of the components that are active in the digital twins, are created considering the organization of the machine. The 3D models are then stored in a model repository and streamed when constructing a visual digital twin in the cloud. Storing the 3D models of machine components brings reusability for a specific type of machine's digital twin creation. Second, when constructing digital twin programmatically, associate specific virtualized operations to specific machines or components.

2.4.1 Granular level assembling of 3D models to form digital twins using MTComm ontology

Digital-twin formation requires breaking down of machine components according to the machine ontology specified in the MTComm probe document. Similar to the XML shown in Figure 20 of MTComm based machine operation where URL of the 3D model of the specific machine component is given to download and construct digital twin. Constructing a 3D visual digital twin using MTComm ontology is a top-down approach. Figure 21 shows a breakdown of a Bukito 3D printer ontology and its associated 3D models for constructing a visual digital twin. CAD models of the machine or its components are associated with the dotted lines. A model renderer module developed in Java is developed to accumulate 3D models of the machine components, shown in the sequence diagram of Figure 22.

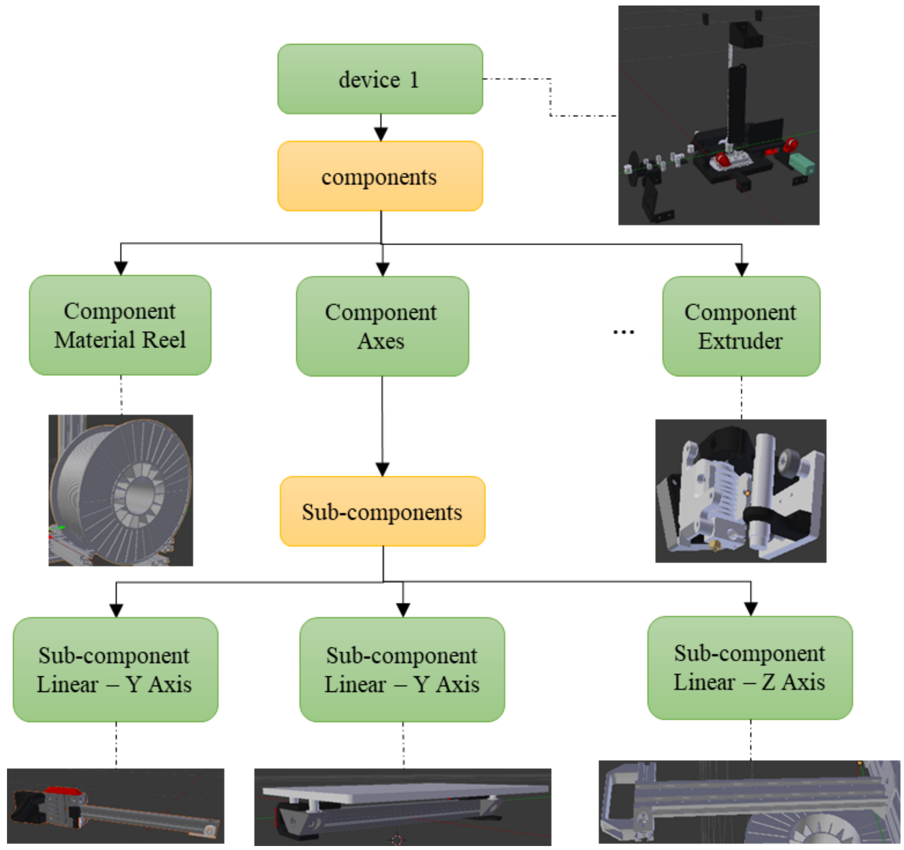


Figure 21: A breakdown of a Bukito 3D printer machine following MTComm based ontology information.

2.4.2 Incorporating manufacturing operations into 3D visual digital-twin components

After the digital-twin assembly is done, a manufacturing operation integration module is developed to incorporate virtualized operations at machine level or its component level. This module also makes sure that during interaction in the GUI of the digital twin, proper actions/events are triggered. For example, when a power switch in the 3D visual digital twin is clicked (in browser-based applications) or touched (mobile based applications), this will change the state of electrical power supply to the machine as a triggered action. The MTComm original method is slightly modified in this experiment too add necessary information of the component CAD models. Figure 22 illustrates the process of assembling a digital-twin and its virtual manufacturing operation integration.

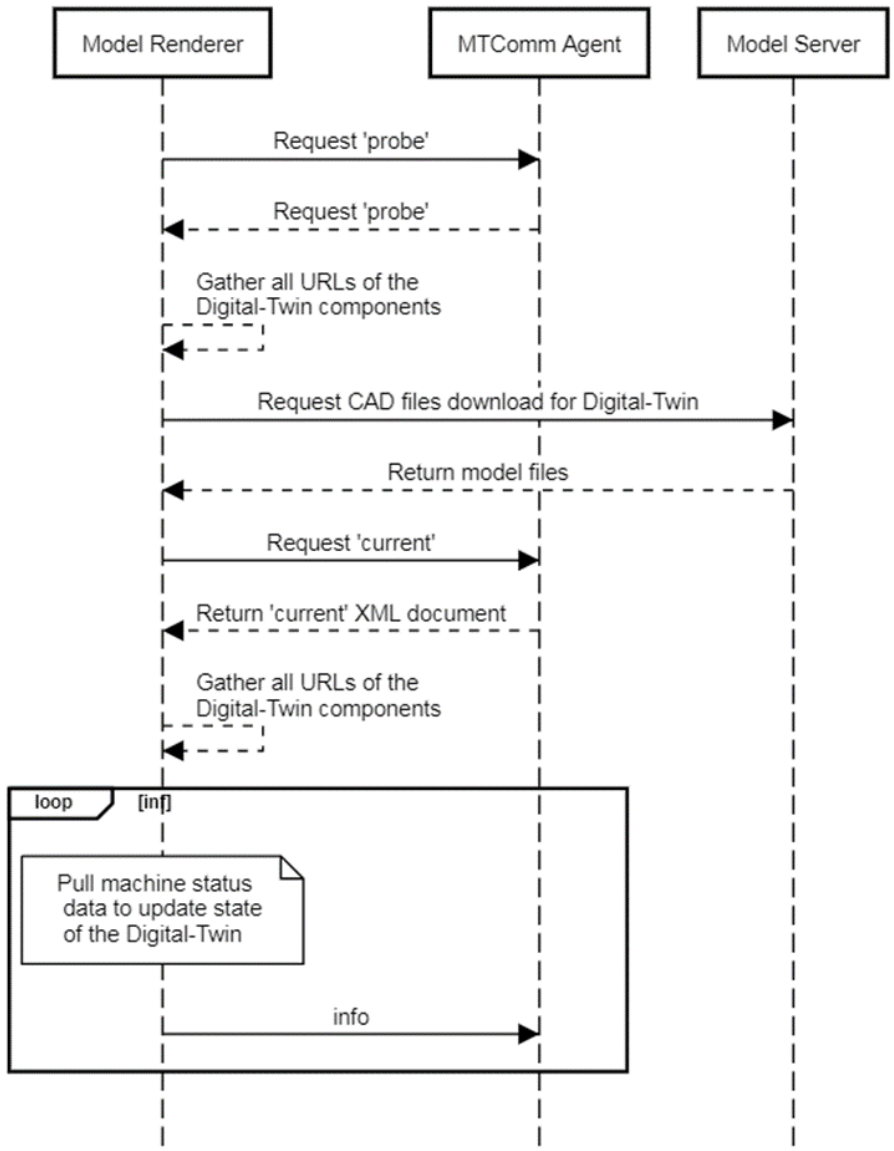


Figure 22: A component level sequence diagram illustrating the process of digital-twin construction.

2.5 Evaluation using manufacturing application scenarios

The developed testbed of CPMC works as a fully functional prototype (similar structure as it is presented in Figure 7). It has three manufacturing sites connected to the cloud over the Internet. Each site has a local area network of machine tools. The first two are inside the University of Arkansas area and the third is inside the Missouri University of Science and Technology area. The two manufacturing sites in the University of Arkansas have six machines in total, including one CNC, three 3-D printers, and two robotic arms, and four controllers, and two private cloud servers. The site in the Missouri University of Science and Technology has one 3-D printer, one robotic arm, two controllers, and one private cloud server.

2.5.1 Application Scenario 1: Components movement simulation during manufacturing operation in the 3D visual digital twin of Bukito 3D printer

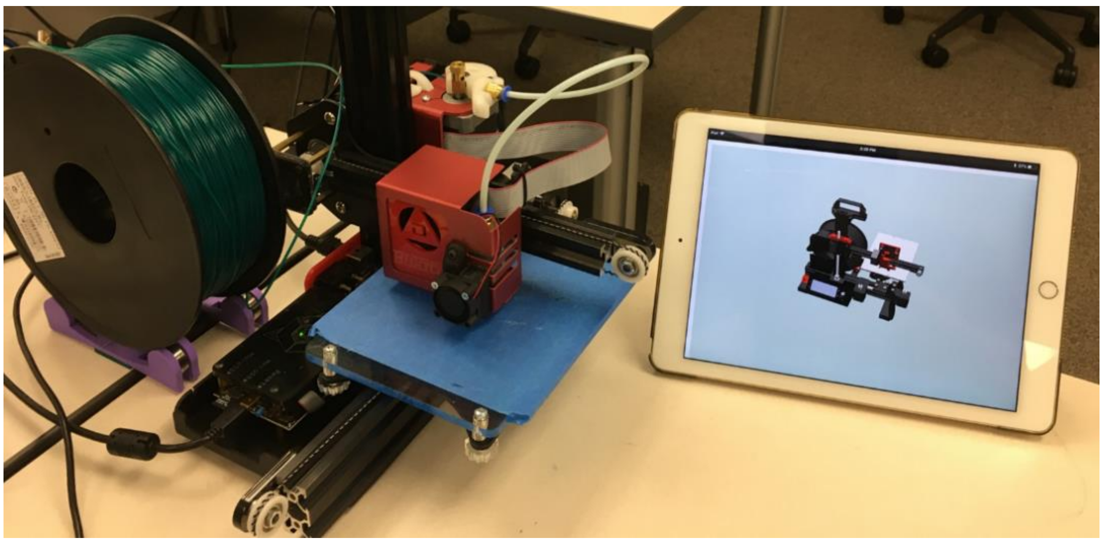


Figure 23: 3D visualized digital twin of Bukito 3D printer rendered on an iPad application

Digital-Twin application simulates the movement of machine components during manufacturing operations. In order to accomplish the simulation efficiently, machine status information is gathered in a frequency of 50Hz. Machine status information is gathered using the MTComm current request. A picture was taken during a manufacturing operation (Printing) in the Bukito 3D printer in the University of Arkansas manufacturing site in Figure 23. The picture shows the Digital-Twin is loaded in an iPad device mimicking machine component movement.

2.5.2 Application Scenario 2: Low filament notification by symbolic change in 3D visual digital twin of Bukito 3D printer

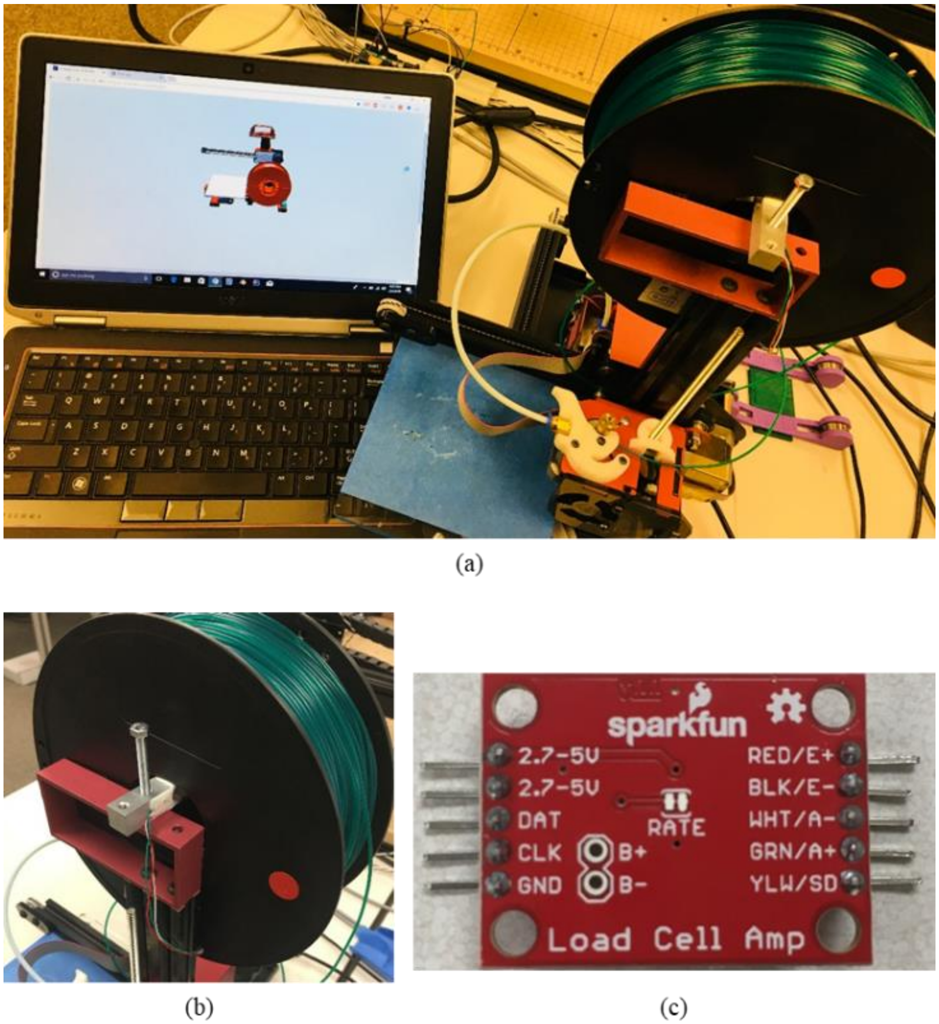


Figure 24: (a) Filament reel in Digital-Twin changes its color to red while weight of the filament reel is below a threshold value; (b) Load Cell setup to measure reel weight; (c) Amplifier for measuring weight of the filament reel.

Another application scenario is implemented in the Digital Twin application notifying the users about low filament for printing. This implemented feature is like low ink notification we get from traditional inkjet printers. When the ink in the printer cartridge is low, it gives a notification to the user showing information about the dying cartridge. Similarly, in the Digital-Twin, the filament reel goes wrong when weight of it goes below a manually set threshold value. Figure 24 shows an overall experimentation of this Digital-Twin feature. The 3D printing filament reel is hanged on a load cell stick in the Bukito machine. A load cell amplifier Sparkfun is used to measure the filament reel weight.

2.5.3 Application Scenario 3: Ensuring maintenance safety by changing colors of heated components in a Digital-Twin

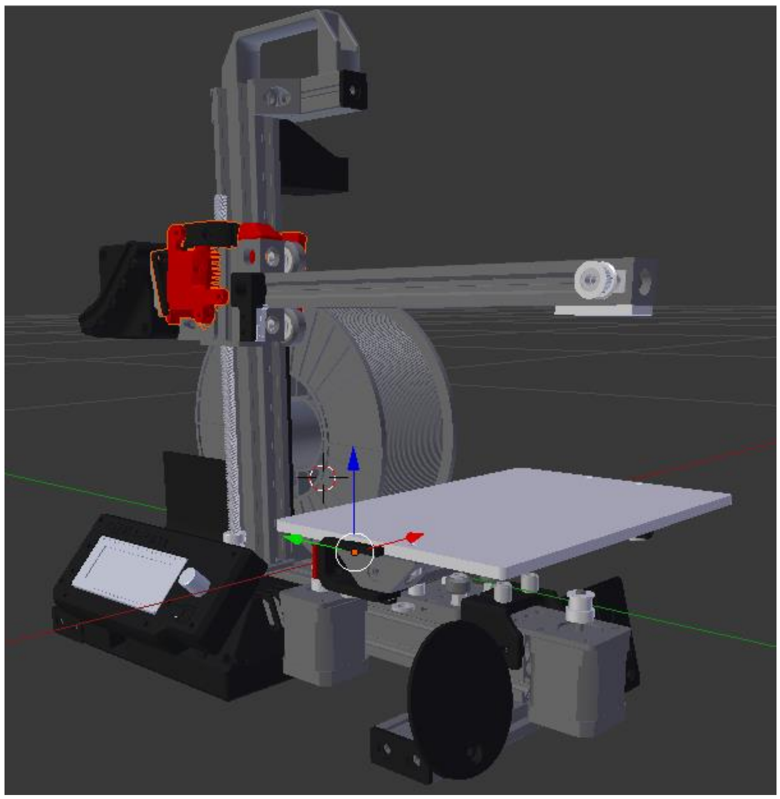


Figure 25: The hot end carriage of the Bukito machine is changed to Red when its temperature is above 200°F degrees Fahrenheit.

During manufacturing operations, machine tools often require heating some of the components. Due to maintenance, those components can cause damage to workers or other surroundings. A Digital-Twin in CPMC can reduce the possibility of damage by showing the proper visualization of those heated components so that workers should use proper precautions to work. In a Bukito 3D printer, the extruder is heated between 200°F-220°F. Touching the extruder before cooling down can cause physical damages to the workers. Inspired to solve this issue, color of the extruder is variated according to temperature change. Workers before starting maintenance the machine, can look at the Digital-Twin, know the status of the heated components, and be careful when handling. Figure 25 shows the extruder is turned red while the temperature was above 200°F and touching it can burn.

2.5.4 Application Scenario 4: Move axes of Bukito machine from the Digital-Twin

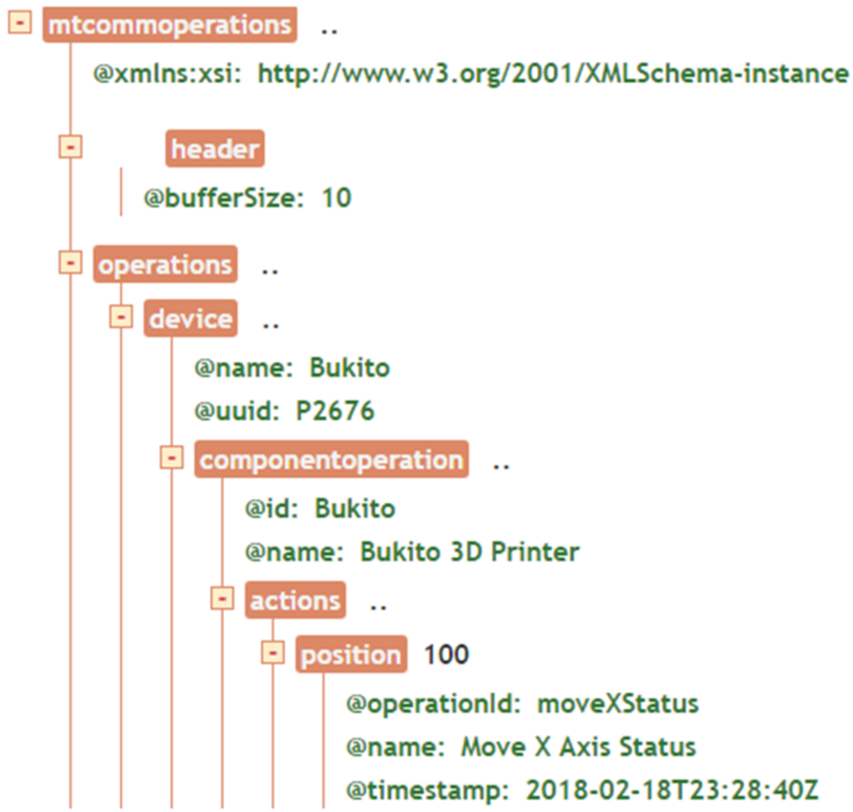


Figure 26: MTComm standard operation XML that sets X-Axis value to 100 of Bukito machine.

Because of MTComm's bi-directional data communication capability, manufacturing operations can be performed right from the Digital-Twin. A feature is implemented in the experimental Digital-Twin, where Linear Axes components can be remotely operated by setting axes co-ordinates. This feature integrates the Digital-Twin and physical machines more tightly coupled both for monitoring and operating. Figure 26 shows the Digital-Twin generated MTComm XML message sent to the machine to perform the x axis movement operation. Presented XML document carries the value of the X-axis co-ordinate. This way the co-ordinate values immediately reflects in the physical machine from Bukito Digital-Twin.

2.5.5 Application Scenario 5: Operate electrical power supply of Bukito machine from the Digital-Twin

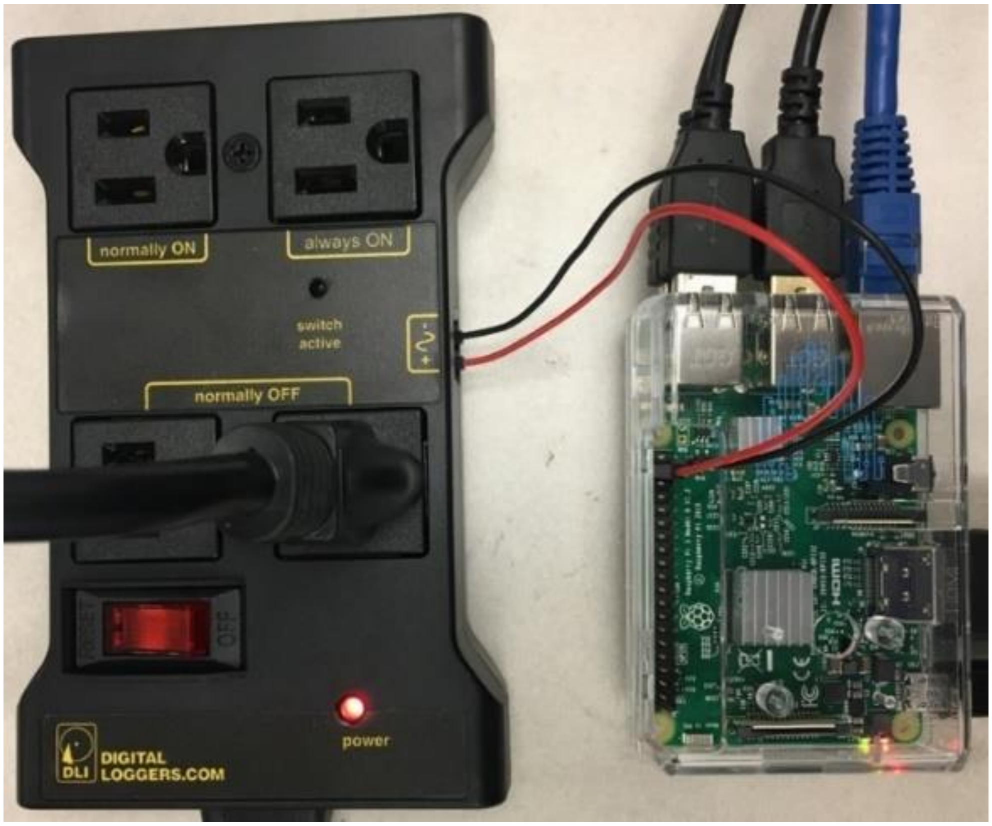


Figure 27: IoT Relay is utilized to control electrical power of the Bukito machine through an MTComm operation service.

To optimize machine runtime and electrical power usage, this manufacturing scenario is implemented. In this scenario, an IOT relay is used to turn on/off the Bukito 3D printer machine over the Internet. Figure 27 gives a picture of the utilized IOT relay setup with a Raspberry Pi to control electrical supply of Bukito machine using a Python script.

2.6 Results discussion

2.7 Delay analysis of 3D rendering during runtime of Digital Twins in CPMC

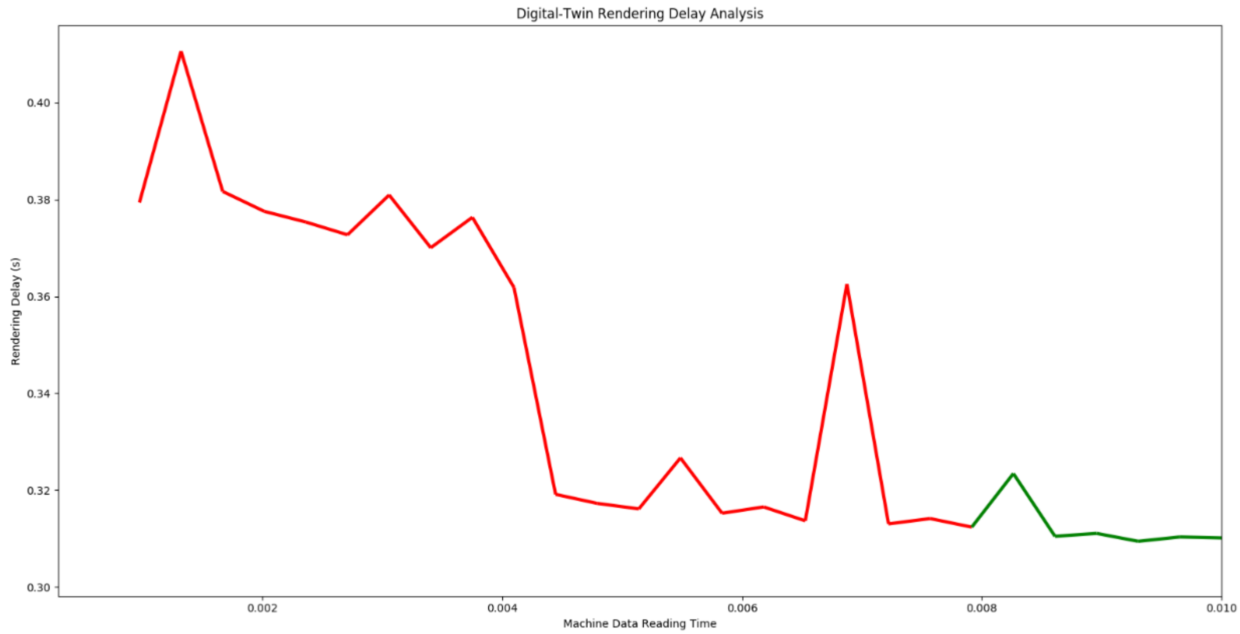


Figure 28: Graph showing network delays on rendering machine movements into Digital-Twin 3D visualization. Red line in indicates a busy machine while green indicates the opposite.

Network delay is significantly influential on the performance of cloud-based digital twins. By digital twins, we get live streaming of the machines right in our gadgets. To evaluate live streaming performance of the implemented digital twins of CPMC, network delay analysis is done by forming a dataset of machines and digital twins. In the dataset, data item reading time of the machines and rendering time of the data items in 3D visual model is collected including other machine status information. Network delays are calculated by subtracting machine reading time from the Digital-Twin rendering time. A graph in Figure 28 is generated from the dataset. Dataset is gathered from different work environment, such as on campus network of the University of Arkansas, home environments, and cellular networks. Network delays are also calculated on several times of the day so that performance can be tested on different network traffic. Collected datasets show that network delays vary randomly from 100 milliseconds to 6 seconds with an average of 0.38 seconds. Larger network delays are observed during machine start up and termination of manufacturing operations. The substance of this experiment, Bukito 3D printer machine moves the highest number of components during operation start and end process. In the graph, higher spikes on the graph are visible during the two mentioned tasks.

2.7.1 Digital Footprint creation

In this work, both information virtualization and visual virtualization of digital twins are developed to accomplish an improved Digital Footprint of machining tools. Information are acquired in the Machine Status Data Repository component in the core cloud layer. The big data repository is developed using HBase, a Hadoop database. However, this approach of machining big data repository is not new, Atin et al. [8] developed such data repository using MongoDB and they collected data using MTConnect protocol. Data in our system is informative than them because of the inclusion of structured machine operation data from the MTComm method. So, the implemented Digital-Twin method of CPMC provides better utility than the mentioned existing system. Performance of MTComm is close to real-time machine monitoring and operating machining tools. Bi-directional communication using MTComm standard messages cost negligible bandwidth consumptions in comparison to recent network bandwidth capacities of the manufacturing sites. While a camera-based shop floor monitoring and operating system demands high bandwidth internet connection, the implemented Digital Twin of CPMC consumes merely the bandwidth of a text-based chatting application.

2.8 Conclusion

This chapter enables digital twins to operate machine tools on a shop floor remotely from cloud. To bridge communication gap between physical machines and digital twins in cloud, paradigm of cyber-physical manufacturing cloud is utilized. MTComm method is used as the primary manufacturing communication protocol. MTComm based virtualization of manufacturing resources allowed them to be orchestrated from cloud. Virtualized manufacturing services are then incorporated with digital twins at a granular level such that activities of each machine components completely match with the digital twins of CPMC. For experimental purpose, a digital-twin of a Bukito 3D printer is developed which are capable of operating machine operations remotely from the cloud-based applications. It also shows machine activities live by mimicking machine itself or its components movements. Evaluation of the developed digital-twin system is done by several manufacturing scenarios. During experimentation with the manufacturing scenarios, network delays and 3D graphics rendering delays are calculated and analyzed. Data analysis shows that the implemented visualization is close to real-time (average delay 0.38s). Comparison with existing state-of-the-art digital twins clearly shows that the idea of driving machine operations from digital twins in CPMC is novel. Results of the experiments show that the developed digital twin is efficient and well structured. In future, we plan to evaluate and expand the proposed method of digital twin in a complex shop floor.

CHAPTER 3: VIRTUAL PRODUCTION LINES USING DIGITAL TWINS IN CYBER-PHYSICAL MANUFACTURING CLOUD

Abstract - Cyber Physical Manufacturing Cloud (CPMC) integrates the paradigms of Cloud Manufacturing and Cyber Physical Systems. CPMC virtualizes manufacturing resources in cloud as web services and publishes them for direct monitoring and operations over the Internet. One can configure virtual production lines by subscribing these published manufacturing web services. However, a web service cannot represent an entire manufacturing resource. Consequently, a remote manufacturer has to subscribe multiple web services to achieve comprehensive monitoring and operation capability. It is desirable to have a more integral virtualization approach of manufacturing resources in CPMC, which will allow a more flexible reconfiguration of virtual production lines when changes occur in production processes. In this chapter, we conceptualize Digital Twin (DT) based Virtual Production Lines in Cyber Physical Manufacturing Clouds. We present an easier approach to monitor, operate and configure virtual production lines using cloud-based DTs of manufacturing resources in CPMC. Towards this objective, we present a service-oriented architecture of DTs for CPMC. We describe a virtualization method and detail a DT-to-DT collaboration workflow. We design and develop a framework to publish DTs in CPMC and configure virtual production lines using them. The framework also describes two workflows to describe virtual production line configuration and execution. To evaluate the proposed DT architecture and framework, we develop and describe an operational testbed. We demonstrate two use cases to validate our proposed architecture, framework, and methods. Finally, testbed results clearly show efficient and easy monitoring, operation and configuration of DT-based virtual production lines in manufacturing clouds.

3.1 Introduction

Cloud Manufacturing [1] virtualizes manufacturing processes in cloud and enable a manufacturer to use them for remote and distributed manufacturing. Typically, remote manufacturing services of cloud manufacturing systems keep human-in-the-loop for operating the machine tools. Cyber Physical Manufacturing Cloud (CPMC) is one of the newly developed manufacturing clouds that integrates the paradigms of Cloud Manufacturing and Cyber Physical Systems. It virtualizes physical machine tools using web services and establish direct monitoring and operations over the Internet [51, 47]. In the contexts of cyber physical manufacturing clouds, one can subscribe these web services and configure them to create Virtual Production Lines (VPL) [61]. However, we identify that web service based VPL are inefficient to manage. Because, subscription to one manufacturing web service allow a user to access one functionality of a manufacturing resource. To achieve comprehensive access to a manufacturing resource, a user needs to subscribe many web services, which leads to difficulty in monitoring and operations of VPL. From another point of view, VPL in modern manufacturing demands flexibility and decentralization when configuring manufacturing processes [61, 62]. It is difficult for remote manufacturing stakeholders to understand complicated physical shop floor situations solely based on web services. In addition, when there is a change in production plans, it is difficult to reconfigure a VPL that has many web services. Therefore, it is desirable to investigate an integral and compact virtualization approach by which a remote manufacturer can get more insightful access to manufacturing resources.

Our objective in this research is to mitigate these problems by conceptualizing a new Digital Twin based VPL in Cyber Physical Manufacturing Clouds. Towards the objective, we design a service-oriented architecture of Digital Twins (DT) that virtualizes physical manufacturing resources in CPMC. Proposed DT architecture offers a more compact and integral service-oriented virtual representations of manufacturing resources than existing service-oriented virtualization techniques in CPMC and other manufacturing clouds [7]. Therefore, it offers easier monitoring and operations of manufacturing resources from cloud. We present a DT-to-DT collaboration method for manufacturing clouds that establishes concurrent collaborations between physical manufacturing resources in shop floors. Inspired from this DT-based virtualization, we propose a novel framework to easily configure, monitor and operate Virtual Production Lines using DTs in CPMC.

Contemporary researchers have aggressively proposed different designs, models and enabler techniques of DTs [63]. DTs are becoming a popular tool for virtualizing manufacturing resources in cyberspace. Requirements in our objective DT design includes monitoring, operating and collaboration capabilities from cloud. A previous study in [64] proposed a cloud-based DT model for monitoring physical machines. [65] proposed a DT model in CPMC that integrated physical machine operations. Researchers proposed multiple DT models, which can monitor, operate and establish DT-based collaborations inside cyber physical production floors [22, 58]. However, these DT models are not designed and suitable for Cyber Physical Manufacturing Clouds. Our proposed DT architecture directly connects manufacturing resources from cloud. We describe a virtualization method based on MTComm [66] based ontology model. Proposed virtualization method generates high-fidelity concurrent simulation of manufacturing resources and enables users to manipulate the machines from virtual representations. It also produces compact virtual representations of manufacturing resources, so that one can subscribe all services of a resource by subscribing a DT. We also describe a DT-to-DT collaboration method and associated information models in the context of CPMC. The proposed collaboration method is detailed in a context of event-driven [67] collaborative manufacturing process established by two DTs of CPMC.

Some researchers [58, 68, 69] proposed DT-based virtual configuration of manufacturing processes. However, their frameworks and methods are limited to cyber physical production floors and not scalable to a global scope as in the contexts of CPMC. We present a VPL configuration, monitoring, and operation framework where DTs are considered as unit elements. Before describing the framework, we conceptualize a model of DT-based VPL and addresses different manufacturing configurations that can be generated from this conceptual model. This conceptual model enables the users to configure VPL, which has manufacturing resources from geographically distributed shop floors. In VPL framework design, we detail the fundamental components to support the proposed concept. Shop floor owners can use this framework to virtualize their manufacturing resources as DTs and publish them in manufacturing clouds. A remote manufacturer can subscribe these published DTs and configure VPL of different configurations. We formalize this VPL configuration workflow using a swim lane diagram. A VPL configuration data model is designed to support the framework and discuss which manufacturing configurations can potentially be produced using this model. In addition, a swim lane diagram is presented to formalize manufacturing process execution workflow based on VPL configurations.

In this chapter, we demonstrate an operational testbed to validate the developed architecture, methods and framework. It consists of five machines separated into two shop floors. Each of the machine is virtualized as a DT in CPMC. We detail testbed implementation that includes overall testbed structure, concurrent 3D visualization using CAD based models, virtualization of physical operations in DTs, and collaboration information design for DTs. Researches in [8, 16] describe different configurations of VPL in manufacturing clouds. We apply two use cases for VPL configurations in testbed. First use case demonstrates a scenario where a manufacturer configures a discrete and parallel VPL using two 3D printing DTs from two shop floors. Second use case illustrates an event-driven collaborative manufacturing process in a VPL where the three machines of shop floor 2 collaborates with each other. Experiments of this prototype implementation validate our research contributions.

This work has the following four research contributions.

- Design of a service-oriented Digital Twin architecture for Cyber-Physical Manufacturing Clouds (CPMC).
- Develop a DT-to-DT collaboration method and associated information models for CPMC.
- A framework to easily configure, monitor, and operate DT-based Virtual Production Lines in CPMC.
- Implementation of an operational testbed for the proposed framework.

Rest of the chapter is organized as follows. *Section 2* presents a literature survey on related topics and describes a background on the previous publications that underpins the concept, methods, and framework of this work. *Section 3* describes the research issues and our objectives for this research. *Section 4* presents a service-oriented architecture of DTs, it's virtualization method, and a DT-to-DT collaboration method. *Section 4* conceptualizes DT-based VPL and presents the framework design. *Section 5* describes a functional testbed implementation to validate the framework. *Section 7* presents a brief feature wise comparison of our work with state-of-the-art and contrast the differences. It also identifies limitations and future work motivations. Finally, *Section 8* concludes the research.

3.2 Literature Survey

3.2.1 Virtual Production Lines in Cloud Manufacturing

Cloud Manufacturing (CMfg) was introduced in 2010 [1] to establish the concept of manufacturing as a service. It integrates Cloud Computing technologies such as service-oriented architectures with industrial manufacturing systems. There are numerous architectures of cloud computing was proposed. Tao et el [7] proposed a computing and service-oriented manufacturing model describing a ten-layer CMfg architecture incorporating the regular manufacturing processes and applications. The architecture also provided important insights on the manufacturing machine virtualization. Researchers took advantage of service-oriented architectures of CMfg and developed numerous models and methods to optimize production processes. Coronado et al. explained the development and implementation of a new manufacturing execution system using MTConnect. They developed Android applications that fetches MTConnect data streams from machine tools to optimize manufacturing production in a virtual factory [70].

As different aspects of industrial manufacturing started to virtualize in cloud manufacturing, manufacturing production lines were also included in the list. Siderska defined a concept of Virtual Production Lines (VPL) as – "a division of a complex creative process into more or less precisely described tasks (jobs) combined with modern IT. The division into tasks, as well as the number of tasks, may be changed throughout the process by actions of experts involved in it. Such a modification is called a self-organization of the VPL." [61]. This author also emphasized that VPL in CMfg creates a scope of sharing manufacturing resources for industrial organizations. We see evidence that large companies are rapidly embracing and applying the concept of flexible and reconfigurable manufacturing with their traditional production sites [71]. Kibria and McLean from NIST presented a virtual reality simulation model of a physical production line [72]. They also demonstrated the difficulties of using simulation modeling for concurrent graphical simulation of mechanical assembly operations and discrete event analysis of a production process. All these works have brought important scientific improvements on the topic of virtual production lines. However, none of the proposed VPL models integrate direct manufacturing operations into cloud representations of manufacturing processes.

We have seen evidence of utilizing concept of Digital Twins (DT) to configure VPLs. Some literatures considered DTs as the basic element of VPL construction. Zhuang et al. proposed a DT based virtual manufacturing framework that detailed real-time data acquisition, synchronization, shop floor assembly construction, big data driven prediction, and production management [73]. Although their framework is based on DTs, they do not describe the virtualization method to construct Digital Twins from granular level. In addition, they only create shop floor simulation to optimize their VPL.

3.2.2 Virtualizing characteristics and capabilities of manufacturing resources as Digital Twins

The term digital twin was introduced publicly by Grieves in 2002 [56]. In 2011, NASA defined DTs as an integrated multi-physics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin [74, 75]. This definition of DT can be contrasted with another paradigm cloud cyber physical systems (CPCC). National Institute of Science and Technology (NIST) defines CPCC as a system environment that can rapidly build, modify and provision cyber physical systems composed of a set of cloud computing-based sensor, processing, control and data services [76]. In this research, we design a cloud-based digital twin model that considers CPCC as an enabler technology to achieve synchronous virtual replica of physical manufacturing resources. To this objective, our DT model adds several features such as machine ontology, geometric assembly, multi-physics and information models on top of CPCC concept.

Many researchers have proposed several architectures and applications of DTs. Tao et al. designed a DT driven product design, manufacturing and service with the help of big data [57]. Despite making significant research contributions, these literatures do not virtualize manufacturing operations. Kai et al. [77] presented a DT based cyber-physical production reference model for automated smart shop floors. In their testbed, they have integrated RFID sensors and computing devices for remote monitoring and operations. Their smart shop floor model dynamically allocates manufacturing subtasks to different machine tools or their specific parts in an optimized schedule. However, their system does not virtualize machine operations in any manufacturing clouds. Redelinghuys, Krugerm, and Basson designed and developed a six-layer architecture to a reference architecture with aggregation of DTs [78]. Lin and Wang presented a 3D virtual and real synchronous mapping, modeling, and operation control technology for automatic test unit. This automatic test line includes a view based 3D scene switching control technology and the scene control instruction mapping technology driven by the actual execution state [79]. None of these developed DTs directly operates physical machines from manufacturing clouds. Lack of research on incorporating physical machine operations in state-of-the-art Digital Twins are carefully identified in a comprehensive literature survey [80].

Apart from virtualization through DTs, several research articles present methods to virtualize manufacturing resources using web services in manufacturing clouds. Liu et al [26] proposed an approach of virtualizing manufacturing resources by representing machine characteristics and service encapsulation using XML. MTConnect [59] is used for virtualizing machine status information using its RESTful web services [81]. Liu et al. proposed a service repository-based method to virtualize machine operations [51]. Several commercial organizations offer DT driven production systems. Oracle Inc. offers a framework that creates three different types of Digital Twins such as Virtual Twin, Predictive Twin, and Twin Projections [82]. Virtual Twin creates digital footprints after production. Predictive Twin collects data and simulates to understand production behavior in the shop floors. Twin Projection offer Digital Thread creation within their cloud. However, their Digital Twins do not execute physical operations from cloud, neither their Digital Twins virtually collaborate with each other. ANSYS Inc. offers physics based Digital Twin simulation software for complex manufacturing resources [83]. However, their DTs are for simulations only, do not incorporate physical operations. Another manufacturing giant Bosch GmbH has developed a DT creation framework named Eclipse Ditto [84]. Eclipse Ditto provides a framework that enables users to work with and manage the state of DTs where the machining data are collected from IoT devices. Our proposed DT architecture in this chapter utilizes MTComm based machine ontology information to virtualize physical machines in manufacturing clouds.

3.2.3 Cloud-based virtual manufacturing collaborations

Cloud-based collaboration of manufacturing services is finding numerous applications in manufacturing domains. Many researches have proposed manufacturing collaboration systems based on service computing technologies. Lu & Xu [67] presented a manufacturing service composition method that facilitates easy mapping between service requests and manufacturing resources based on restrictive rule sets in the cloud and availability information about a resource. Service-oriented manufacturing services enable remote collaboration of manufacturing resources in cloud. Yang et al. presented a Large Equipment Complete Service collaborative logical framework based on cloud manufacturing service platform [85]. This framework describes a platform to collaborate with enterprise external manufacturing services in cloud. Valilai and

Houshmand introduced XMLAYMOD that supports cost-effective distributed manufacturing collaboration through manufacturing cloud services [86].

All these researches provided significant insights on compositing virtualized manufacturing services in the cloud and possibly achieving manufacturing collaborations. In the context of CPMC, a web service based collaborative manufacturing method was presented in [49]. However, DT-to-DT collaborations architecture has a different set of features than traditional web service-based collaboration approaches in manufacturing clouds. DTs are autonomous representations of a manufacturing resources. They can assess different situations during collaborations more intelligently and take decisions autonomously. On the other hand, traditional service-based collaboration methods typically rely on centralized Agent modules. In addition, DTs offer better collaboration process monitoring approaches using data driven and visual aids, which is difficult to achieve in existing web service-based approaches. Therefore, DT-to-DT collaborations can potentially lead to an easier to monitor, more flexible, self-adaptive and decentralized collaborations in manufacturing clouds. In this chapter, we investigate and present a DT architecture that establishes DT-to-DT collaborations in CPMC. We also describe the roles of DT-to-DT collaborations in virtual production lines.

3.3 Research issues and objectives

The following are motivational research issues for this research.

- **Representation of manufacturing resources and processes in manufacturing clouds:** The approach in Cloud Manufacturing systems is to virtualize manufacturing processes as services, so that stakeholders can access them in a global scope. In CPMC, the approach is to directly virtualize manufacturing resources in a global scope by cyber physical system integration and service-oriented architecture. Both of these approaches are cloud-based but differs in virtualization approaches. One can subscribe and composite different web services to configure a VPL in CPMC. But this approach leads to difficulty in service management due to numerous web service subscriptions. Organizing these services for monitoring and operating for a virtual production process impedes system efficiency. It is difficult for a remote operator to gain comprehensive operational capability on involved resources. Addressing these issues, we investigate a more objective oriented and better packaged approach to virtualize manufacturing resources. DT has already been gaining momentum in virtualization of manufacturing resources [80]. However, existing DT models for manufacturing clouds do not describe virtual-physical dynamics. On the other hand, DTs proposed for cyber physical production systems does not virtualize a manufacturing resource for global scope. Clearly, there is a research gap in virtualization approach that we investigate in this chapter.

- **Cloud-based architecture DTs:** Many existing DTs virtualize manufacturing resources by collecting machining data but do not allow physical operations from cloud. A few state-of-the-art integrate physical machine operations with DTs in cyber physical production systems. A DT based reconfigurable cyber physical production system is presented in [69]. This study develops DT instances from machine ontology information and uses an internet scale MQTT-REST combination to monitor and operate manufacturing resources. Communication method of this study has two limitations. First, this design is not suitable for persistent DTs of large and complicated manufacturing resources. Secondly, their design of communication method does not describe integration with manufacturing clouds. In addition, they use MQTT with REST protocol to enable internet communication, which lacks directness in remote access. Another contemporary [68] proposes an internet scale control of manufacturing resources in a DT based cyber-physical production systems. Both of these state-of-the-art present internet scale DTs for cyber physical production systems but do not describe them in the context of clouds. Addressing this issue, it is desirable to investigate a cloud-based model of DT with improved virtualization and communication.

- **Configuring virtual production lines using cloud-based DTs:** Modern virtual production lines demand frequent changes in manufacturing processes. Existing studies in [69, 68, 58] enable creating virtual production lines within a cyber physical production system for remote and personalized production. But in the context of cloud manufacturing, a virtual production line can be distributed across shop floors globally [61], which these contemporaries cannot offer. Addressing this issue, we investigate a new framework that can configure distributed virtual production lines across manufacturing organizations. We identify that if manufacturing resources are virtualized as cloud-based DTs under a framework, they become globally accessible. Thus, such framework will have a wider scope of configuring virtual production lines.

Motivated from the above discussion, we develop a concept of DT-based VPL in CPMC. We propose a virtualization method using DTs for manufacturing clouds, which has one-to-one relationship with manufacturing resources. We also investigate how a DT can bundle all manufacturing web services and information models of a manufacturing resource into one globally accessible endpoint in cloud. We envision that shop floor owners will publish DT of manufacturing resources and virtual manufacturers will subscribe them and configure VPL. To this objective, we propose an architecture of cloud-based DTs (DT), which integrates web services for machine status and operations at a granular ontology level. We design and develop a framework to fulfill the vision that offers a closed loop publish-subscribe-configure-monitor-operate DT-based VPL for manufacturing clouds.

3.4 Service-Oriented Architecture of Digital Twins for CPMC

Towards the objective of this research, we present a new design of DTs with the following requirements.

- First, the design must follow a service-oriented architecture (SoA) for manufacturing clouds, so that users can access a DT in a global scope via multiplatform applications.
- Second, the design must offer an integral service package that contains services for data driven monitoring, visualization driven monitoring, virtual operations in a DT, so that users can subscribe an entire manufacturing resource at once.
- Third, the design must enable cloud-based DT-to-DT collaboration, so that users can configure virtual production lines from cloud.

Existing DT models in [58, 69, 68] describe cyber-physical dynamics of DTs in the contexts of cyber physical production systems. Therefore, these models do not satisfy to all of the above requirements for manufacturing clouds. There are existing cloud-based models of DTs. For example, the study [64] present a cloud-based DT model for monitoring FDM based additive manufacturing machines. However, they do not integrate physical machine operations into their DTs. In addition, their architectures do not satisfy the above requirements.

To satisfy the above requirements, we present a DT design in the three following sections. First, we present a SoA of DTs in CPMC. Then, we present a virtualization method to support the SoA. Finally, we develop amethod of DT-to-DT collaborations in CPMC.

3.4.1 Architecture of DTs

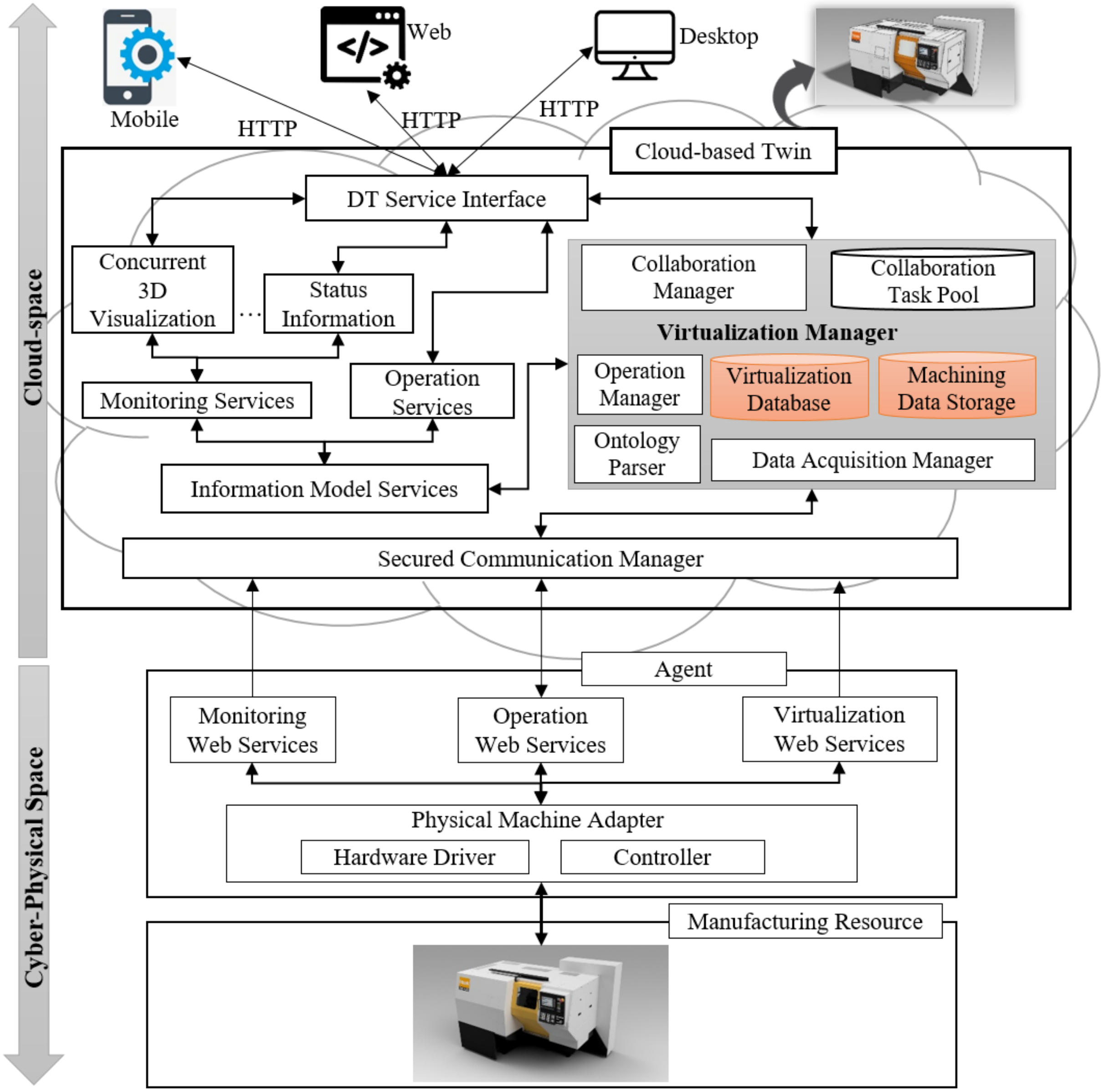


Figure 29: A service-oriented architecture of DT for CPMC. This architecture virtualizes a manufacturing resource and creates a synchronous representation using data driven and visualization driven monitoring. The Cloud Twin executes physical machine operations directly and capable of establishing DT-to-DT collaboration in cloud within a virtual manufacturing process.

A service-oriented architecture of DT is presented in Figure 29. This design consists of three following modules such as a Manufacturing Resource (MR), an Agent and a Cloud Twin. MR consists of one or multiple shop floor machine tools. Agent module integrates cyber capacity on top of the MR. Cloud Twin module creates a service-oriented virtual replica of the MR.

Agent part remains within a local area network of MR. It is hosted in a computing node in shop floor and has the following components. It has a component, namely Physical Machine Adapter, which consists of two sub-components such as a Hardware Driver and a Controller for the MR. The Agent part has a service-oriented architecture and offers several web services using few subcomponents such as Monitoring Web Services (MWS), Operation Web Services (OWS), and Virtualization Endpoints (VWS). MWS provides web services for collecting sensor data. OWS provides web services for executing physical operations from cloud. When virtualizing a physical machine as DT, the VWS subcomponent provide a semantic ontology presentation of the MR. Through these web services, Agent enables a two-directional Internet connection between manufacturing clouds and MR. It also validates any incoming and outgoing communication from cloud.

Cloud Twin module represents the MR as a DT in CPMC with the following features. It offers remote monitoring through Status Information and Concurrent 3D Visualization services. It enables direct physical operations of MR from cloud. It also produces high-fidelity CAD model-based 3D visualization of physical machines. The 3D visualization synchronizes concurrently with the states of the physical machines. When concurrently synchronizing physical machines, it collects machining data through the Agent module. Cloud-based applications access this Cloud Twin module in order to load and render the DTs in an interactive graphical user interface. This interface enables users to directly send commands to MR via the Agent.

Design of Cloud Twin module includes the following components. Secure Communication Manager (SCM) component establishes communication with the physical machines. The Virtualization Manager (VM) component is the heart of Cloud Twin and stores monitoring and operation information of MR. It has multiple subcomponents such as Data Acquisition Manager

(DAM), Ontology Parser (OP), Operation Manager (OM), Virtualization Database (VD), Machining Data Storage (MDS), Collaboration Manager (CM), and Collaboration Task Pool (CTP). OP extracts virtualization information from VWS subcomponent of Agent and stores them in VD. Using information from VD, DAM component collects machine status information through MWS of Agent. OM collects virtualized operations from VD and sends the operation commands to the MR via OWS of Agent. Collaboration Manager (CM) of VM establishes and executes event driven collaboration with another DT in CPMC. An external service in cloud can send a task to collaborate with another DT through the DT Service Interface. The collaboration tasks are initially buffered to Collaboration Task Pool, so that CM can execute them sequentially.

Information Model Services (IMS) is another key component of Cloud Twin module. Benefited by the VM component, it manages and offers different representation of the MR in cloud applications. For monitoring, it interacts with Concurrent 3D Visualization (C3DV) service to render and synchronize the near real-time 3D visualization of MR. IMS cooperates with Status Information service to generate data driven monitoring of MR. It also works with the Operation Services to allow incoming operation requests for MR from cloud applications. All these services for monitoring and operations are organized through an interface, namely DT Service Interface. This fulfills the requirement of a packaged access to DTs from cloud.

3.4.2 Virtualizing manufacturing resources in manufacturing clouds through DTs

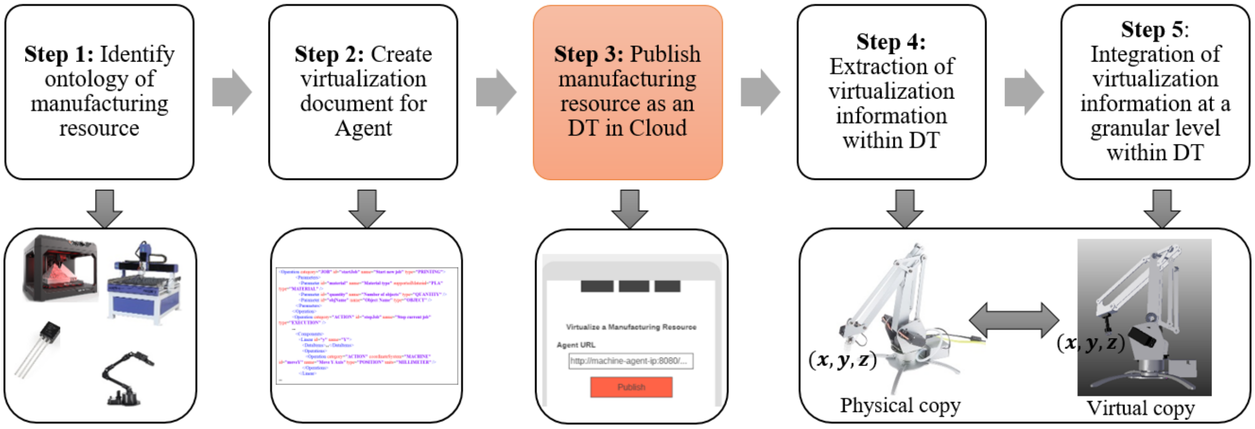


Figure 30: Five steps of virtualization method to construct a DT in CPMC.

Figure 30 presents a common virtualization process of virtualizing MR as DTs. This process consists of five steps. First step identifies a machine ontology information for virtualization. For example, an ontology information model of a 3D printer will contain monitoring data (e.g. axes co-ordinates, bed temperature, extruder temperature, etc.) and physical operation information (e.g. print a 3D model file, start, stop, pause, resume printing etc.). Second step in Figure 30 converts machine ontology into a virtualization document and stored in VWS of Agent. For ontology design, we consider XML structures gathered from MTComm's [66] 'probe' web service as preferred approach. In third step, an MR is published in cloud as a DT using the framework. In step four, VM component (from Figure 29) of DT parses the virtualization document and organizes information in Virtualization Database. Finally, DT embeds machining data and virtualized operations at a granular level with geometry models of the MR at a granular level.

Virtualization methods in our research has similarity in approach with the existing DT models of cyber physical production systems [69]. However, there is a technical improvement in this virtualization method because it uses ontology design and virtualization web services of MTComm method. MTComm provides a specific and generic data structure for directly operating

MR of different scale, which does not exist in state-of-the-art DT models. Below we present a semantic ontology design for MTComm enabled manufacturing resources.

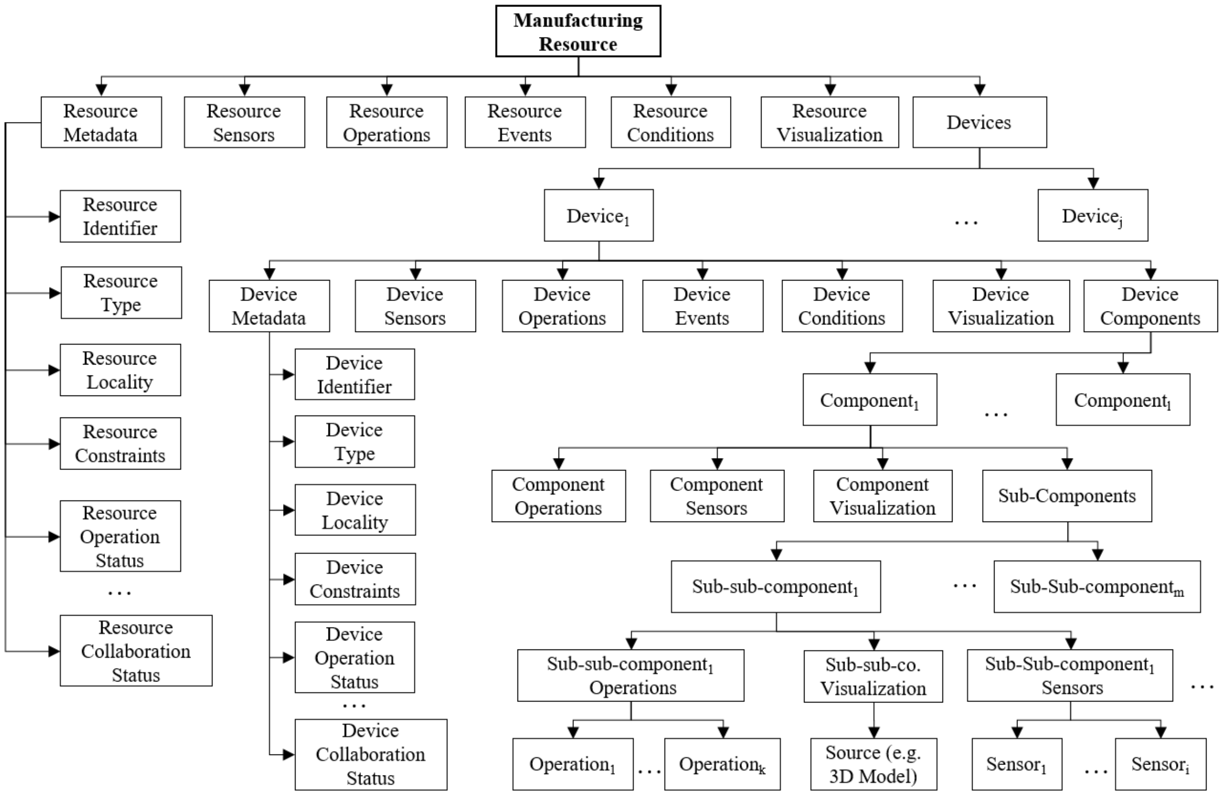


Figure 31: Semantic ontology structure of an MTComm compliant manufacturing resources.

At the top of Figure 31, a manufacturing resource consists of metadata, system level sensors, actuators, and devices. Ontology of each devices are similarly designed. This tree hierarchy expands the data organization from manufacturing resource level to sub-sub-component level in a similar organization. This physical system data organization is followed and modified from the original implementation of MTComm [66]. New additions in this ontology include support for an aggregated 3D visualization and support for DT collaborations. The following presents an example virtualization information of a 3D printing machine based on this semantic ontology structure.

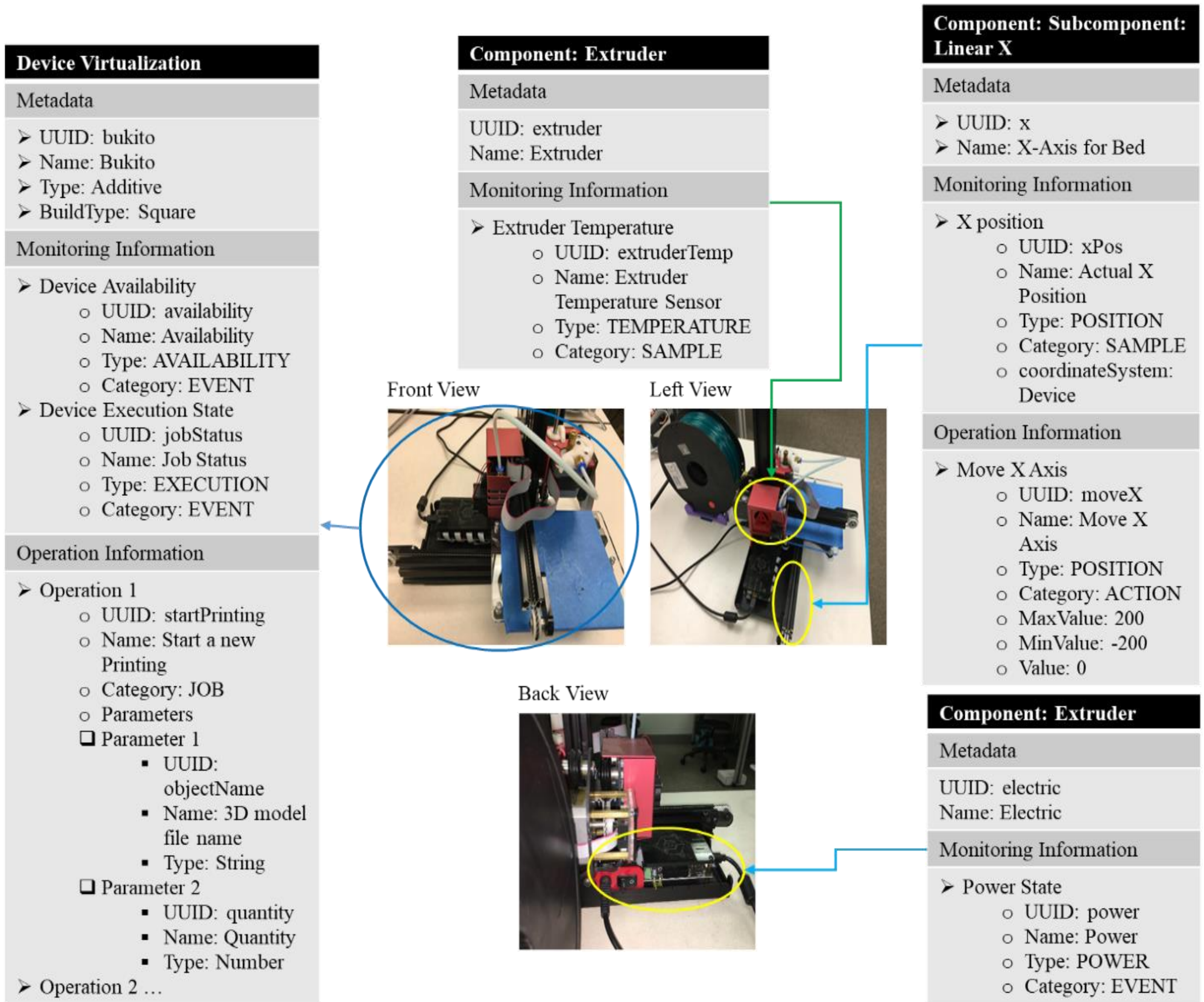


Figure 32: An example virtualization information mapping for a Bukito 3D printer for monitoring and operations in a DT.

Figure 32 presents a sample virtual information mapping for a Bukito 3D printer based on MTComm's ontology. This mapping is achieved after step 4 of Figure 30. This information mapping is partial but covers all mappings for monitoring and operations. There are three pictures of the Bukito 3D printer taken from three different views to focus on the parts. The Front View image focuses on the whole machine and it helps to illustrate the virtualization information at device level. Virtualization information at device level has two monitoring information and multiple operation information. Monitoring data items are about device availability and device execution state. The only virtualized operation information shown in the figure performs a printing job where the parameters take inputs such as name 3D model file and number of printed copies. The Left View and Back View focus on different components and subcomponents of Bukito 3D printer, which help to illustrate the virtualized information for monitoring and operation. The component extruder has a virtualized sensor for temperature recordings and some metadata information about it. Subcomponent Linear X Axis has two virtualized monitoring information such as X position and component operation status. It has a component level operation of type ACTION. This operation can take a numeric value to move the X axis.

3.4.3 DT-to-DT collaboration in CPMC

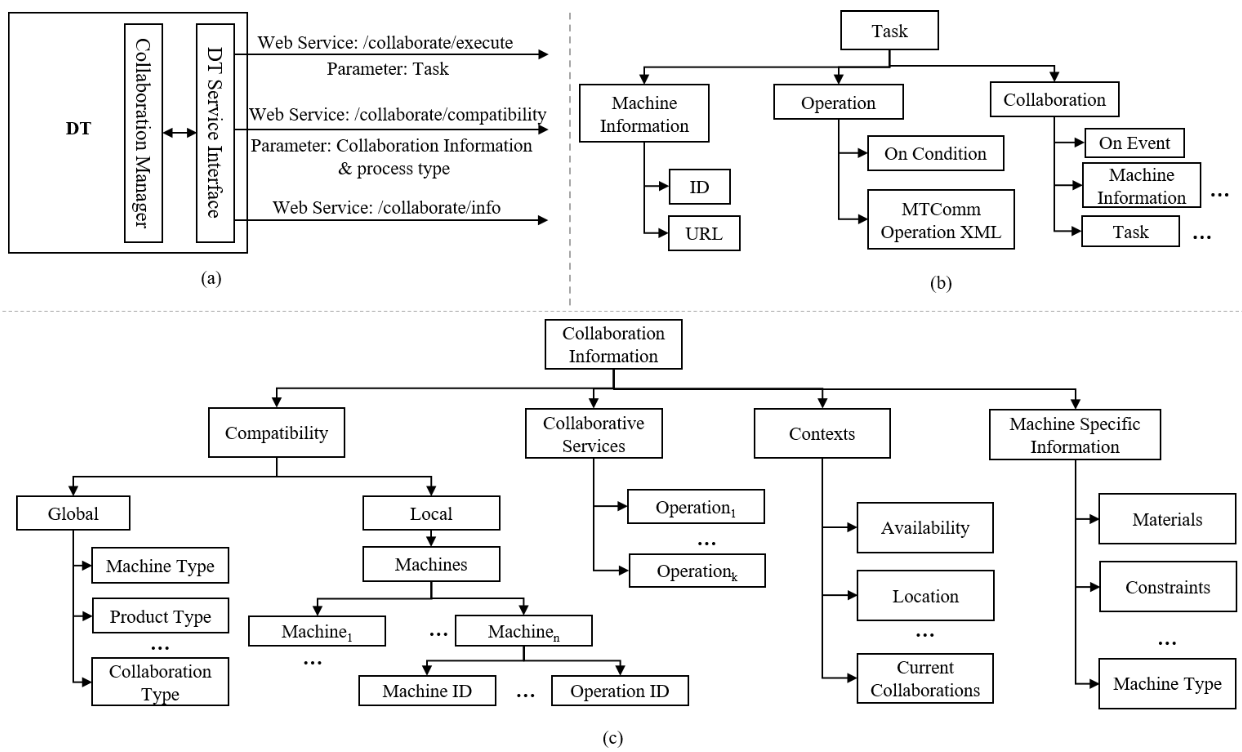


Figure 33: DT-to-DT Collaboration in CPMC. (a) presents web services offered by a DT to establish collaboration; (b) presents a manufacturing task information model for executing a manufacturing operation and invoke another DT for event-driven collaborations; (c) presents a collaboration information model to validate collaboration between two DTs.

Figure 33 (a) presents the web service layout for collaboration in a DT. These services are accessible through the DT Service Interface component. The following are the three web services for collaboration. First, execute web service performs an operation in MR and its parameter is an executable Task. Information model for a Task is presented in Figure 33 (b). A task model includes necessary information for executing an operation while a specific condition satisfies in MR. Collaboration Manager (CM) component drives the execution of a task. A task can include a follow up collaboration task, which the CM executes on another DT. Information model of task follows a linked list data pattern for event-driven production process configurations.

The second web service for collaboration is compatibility, which validates potential collaboration between two DTs (source and target). As functional input, it takes collaboration information of a target DT, for which a model is presented in Figure 33 (c). It has four categories of information. Compatibility information contains two subcategories such as Global and Local because a collaboration can happen locally inside a shop floor or globally across shop floors. For global scope, collaboration information includes machine type, product type, collaboration type etcetera of target DT. For local scope, it includes which specific target machine tools the source DT can collaborate with. Another categorical information is contexts, which includes MR availability, location, enrolled collaborations, etcetera. Machine Specific Information includes characteristics and metadata of the source MR. Last but not the least, Collaborative Services enlist MR operations which are enabled for target DTs to access. Therefore, manufacturing web services of a DT are classified in two categories. Firstly, the virtualized operations that are accessible publicly, particularly from cloud-based applications. Secondly, the virtualized operations, which can be requested from another DT. For example, a Bukito 3D printer has the following operations such as i) 3D print, ii) move X-axis, iii) resume printing, and iv) pause printing. All these operations are published as public web services in the Bukito DT. However, all these operations are published as collaborative except the "move X-axis" operation. When collaborating with DT of a robotic arm, the robotic arm can collaborate for printing operation, but will not request the "move X-axis" operation.

The third web service in Figure 33 (a) is info, which accepts a HTTP GET request and returns the information model in Figure 33 (c). Collaboration Manager keeps the information model of Figure 33 (c) up to date at a real-time basis using other sub-components of Virtualization Manager component.

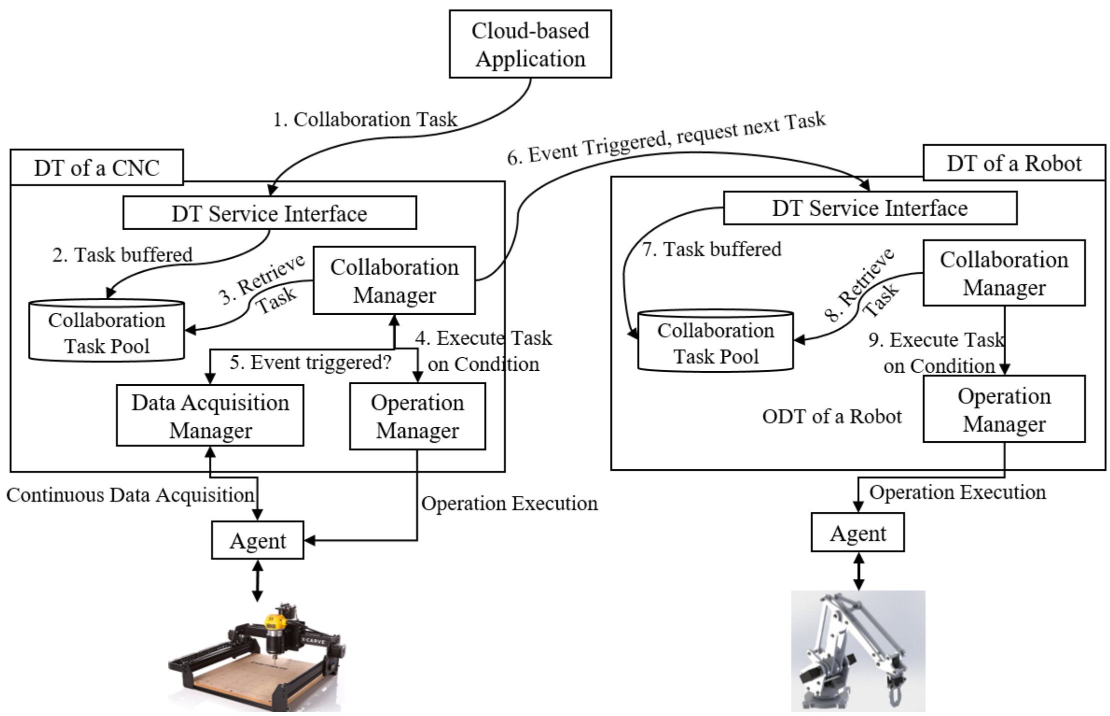


Figure 34: An example workflow showing a CNC DT collaborating with a robotic arm DT in CPMC. This workflow also details how DTs take decision on next task (Figure 7 (b)) in an event driven collaborative manufacturing process.

Figure 34 presents an example collaboration workflow between a DT of a CNC machine and a DT of a robot machine. Each of the steps in this workflow are labeled with incremental sequences and messages. The context of this example is based on execution of an event driven collaborative manufacturing process, which is initiated by a cloud-based application. We assume both the MRs in this example reside within a shop floor and their collaboration is pre-validated before initiating the process. The diagram shows business logics of this collaborative process that initiates and remains in clouds. It also shows how the decision support system of DTs work while executing an event driven collaborative production process.

3.5 Framework for DT-based Virtual Production Lines in CPMC

Contemporaries propose different methods for reconfigurable production lines of manufacturing processes inside a cyber physical shop floor. A study in [58] presents a generic reconfiguration of production lines based on the production orders. They describe a DT-based cyber shop floor where they perform simulation before reconfiguring production lines for new orders. Following a similar strategy, [68] presents a personalized 3D printing cyber-physical production system, where each machines and robots are accessed through their DTs and reconfigure production line based on order. They also present multiple algorithms to detect and mitigate machine failures using the DTs. [69] presents a reconfigurable cyber physical production system based on DTs, where they describe reconfiguring production lines for internal event-driven cases and external web-based changes. It is evident from these promising researches that DT-based reconfigurable production lines are gaining traction in manufacturing domain. However, all of these methods reconfigure cyber physical shop floors. None of them describe a framework to configure virtual production lines in manufacturing clouds. Therefore, their reconfiguration methods are applicable in local scope of a shop floor.

Work of this section adds values to this research in three-fold. This framework publishes manufacturing resources (MR) in CPMC as DTs. It allows users to configure VPLs and their manufacturing processes using DTs instead of individual web services. In addition, enables configuring virtual production lines that can exist locally inside a shop floor or globally across multiple shop floors. Hence, offering a wider scope of distributed VPL in manufacturing clouds.

To present this framework, first we conceptualize reconfigurable VPLs in CPMC. Then we present the framework with workflows of configuration a VPL and execution of associated virtual manufacturing process.

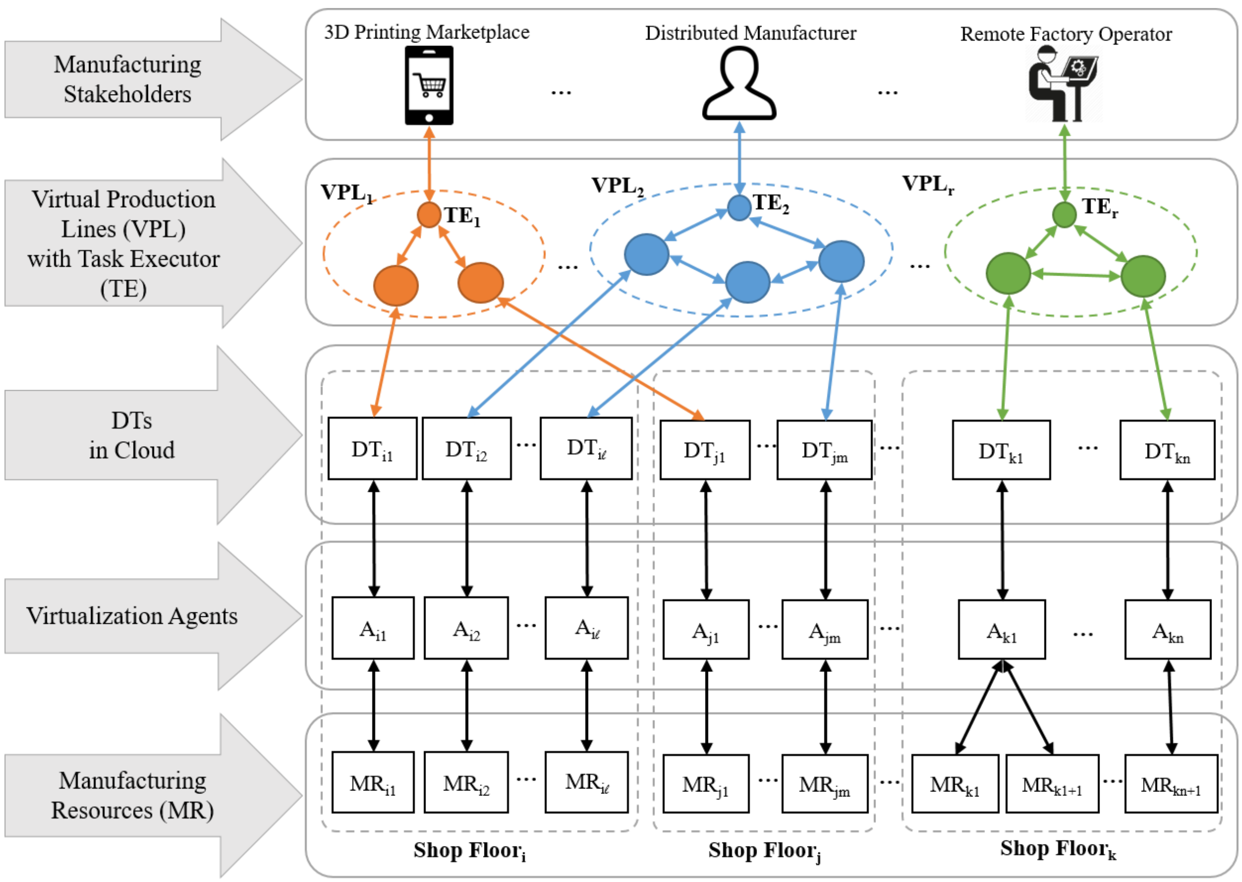


Figure 35: Conceptual diagram of DT-based VPL in CPMC. Each VPL associates a virtual manufacturing process of different configurations showcase high-flexibility and reconfigurability production lines.

3.5.1 Conceptualizing reconfigurable VPLs in CPMC

Figure 35 shows a VPL is constructed by combining more than one DTs for a production goal. From the bottom, MRs in shop floors are connected to virtualization Agents. An Agent can connect with one or multiple MRs. Each of these MRs are then virtualized as cloud-based DTs.

One DT connects to its MR twin via one Agent. Therefore, a DT can represent one or more MRs in physical shop floors. We assume that DTs are constructed and hosted in cloud-based containers such as Docker instances. After construction, DTs are published in manufacturing clouds for users to access them publicly to configure VPLs.

Conceptual diagram shows multiple example VPLs created using the published DTs. VPL_1 consists of two DTs for a discrete manufacturing process and capable of parallel processing; VPL_2 presents an event-driven collaborative manufacturing configuration using three DTs; and VPL_r consists of two DTs of three manufacturing resources and shows a highly flexible manufacturing process. Among these three VPLs, VPL_1 and VPL_2 create globally distributed manufacturing process. Contrarily, VPL_r outputs a locally reconfigurable production line.

Each of these VPLs are operated by a Task Executor (TE), which take execution order from different manufacturing stakeholders. VPL_1 is suitable for a 3D printing marketplace where one can rent MRs across shop floors and create a discrete manufacturing process. VPL_2 is suitable for a distributed manufacturer, who executes manufacturing orders based on information exchanges or physical collaborations between DTs. VPL_3 can be useful for remote manufacturing operators to configure production lines on the basis of demands. This conceptual model can potentially support numerous manufacturing configurations due to its flexible DT-based compositions.

3.5.2 Framework for configuring, monitoring and operating DT-based VPLs in CPMC

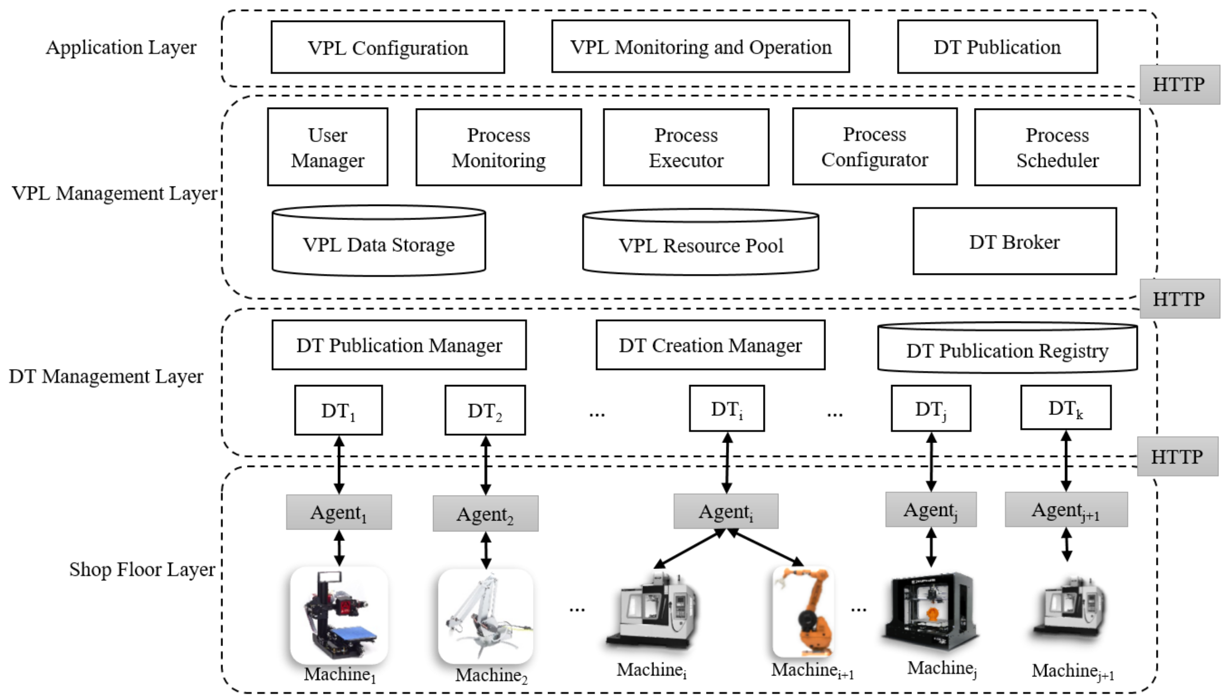


Figure 36: Framework design of DT-based VPL in CPMC. This framework has four layers as in CPMC, except the functional components are different. This framework employs HTTP based communications in all layers.

Figure 36 presents a design of a framework for configuring, monitoring and operating DT-based VPLs in CPMC. Components and resources of this framework are distributed in four layers. At the bottom, Shop Floor Layer consists of physical shop floors where MRs use service-oriented Agents to enable communication from manufacturing cloud.

DT Management Layer contains the DTs and several other components to manage them. DT Creation Manager component helps creating DTs in CPMC. A detailed work routine of this component is presented in a previous research [65]. DT Publication Manager component contains a cloud-based application that allows users to publish their created DTs in manufacturing clouds. When publishing a DT, this component registers DT address, metadata, collaboration and ontology information into a database, namely DT Publication Registry. This database helps searching suitable DTs for a user.

VPL Management Layer consists of multiple components and data stores. User Manager component maintains user subscription and protects access to VPLs. Process Configurator component provides web services to configure VPL using DTs. Process Monitoring component offers web services to access data driven and visualization driven monitoring of VPL. Web services in this component aggregates monitoring web services from all involved DTs. Process Scheduler component buffers manufacturing execution requests from cloud-based applications in a queue. Process Executor component polls the queue of Process Scheduler component and executes the manufacturing tasks in respective VPLs. VPL Data Storage component stores manufacturing process monitoring and operations data of VPL. Note that, we already have a data storage in DTs. The objective behind using a different data storage for DTs and VPL is to isolate MR data and production process data in manufacturing clouds. VPL Resource Pool component is a data store that persists the configured VPL and allow other components and applications to retrieve instances on request. When configuring a VPL, DT Broker component facilitates the workflow by recommending suitable list of DTs sorted by compatibility score. Compatibility score for a pair of DTs is calculated based on their match cases in collaboration information. Note that, compatibility score is useful when a VPL has an event-driven collaborative manufacturing configuration.

The Application Layer contains cloud-based applications for configuring and administering VPL in CPMC. We show three example applications in this layer to describe the fundamental ideas of this framework. VPL Configuration application help a user to configure VPL. A user can monitor a VPL through data and 3D visualization using the VPL Monitoring and Operation application. They can also execute manufacturing process associated with VPL. A shop floor owner can publish DTs of his MRs using DT Publication. For cloud support, scalability and flexibility, all communication of this framework is conducted using HTTP based web services.

The followings describe decision support systems of this framework when configuring and executing manufacturing processes of VPL.

3.5.2.1 *VPL configuration workflow*

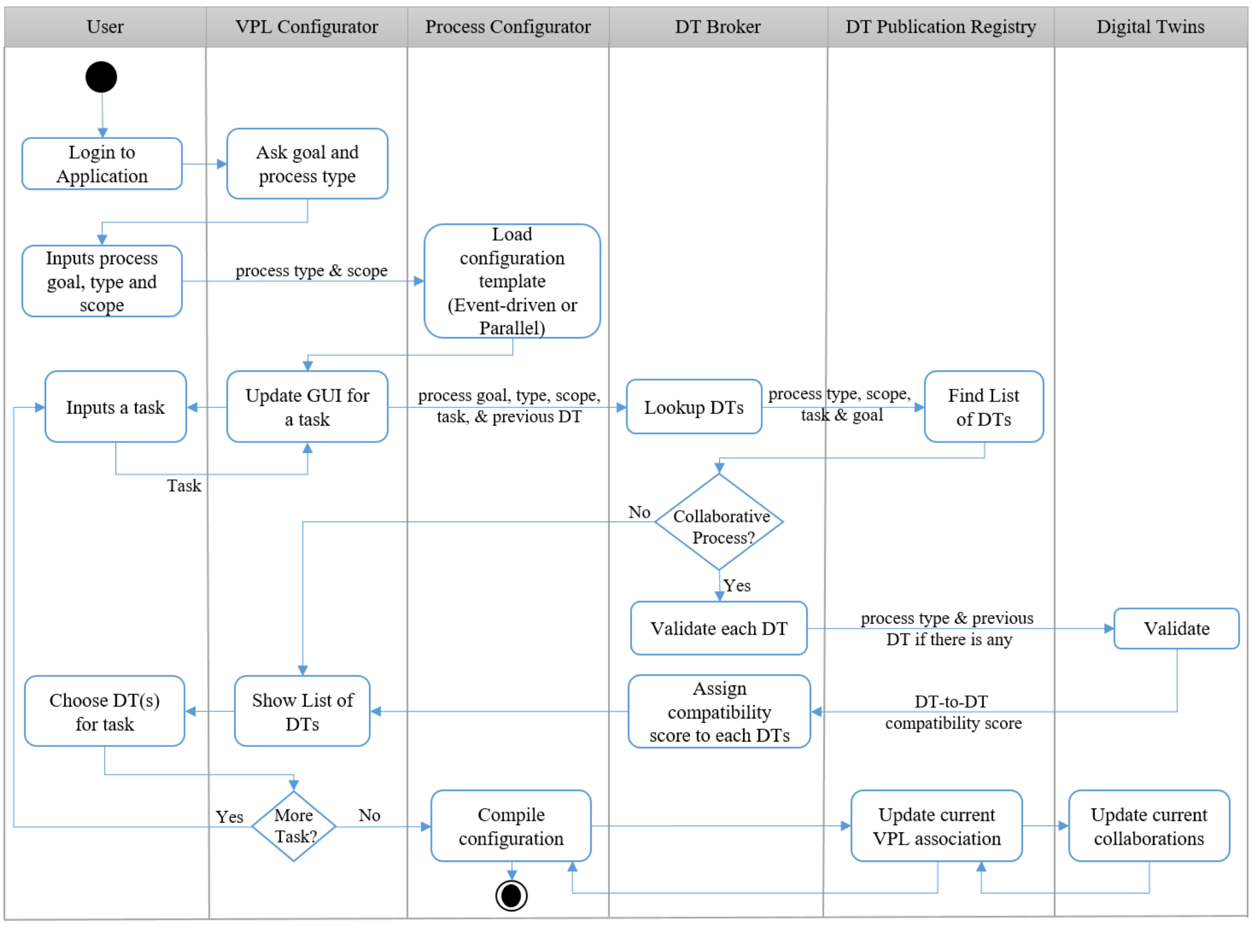


Figure 37: Swimlane diagram to present DT-based VPL configuration workflow in CPMC. Generated configurations can support different manufacturing processes such as discrete, event-driven, collaborative, sequential, parallel etc.

Figure 37 presents a swimlane diagram to demonstrate DT-based VPL configuration workflow in the proposed framework. This entire workflow remains inside the cloud-based components where they interact with each other. The workflow starts from VPL Configurator application, where a user logs in and inputs information about a manufacturing process. Based on the user inputs, VPL Configurator application commands the Process Configurator component to help the users to choose suitable list DTs for manufacturing tasks of a process. DT Broker component recommends suitable DTs by fetching from DT Publication Registry. When collaboration is required, it validates compatibility within the scope of VPL. The diagram details each of the steps and decision points to configure a VPL.

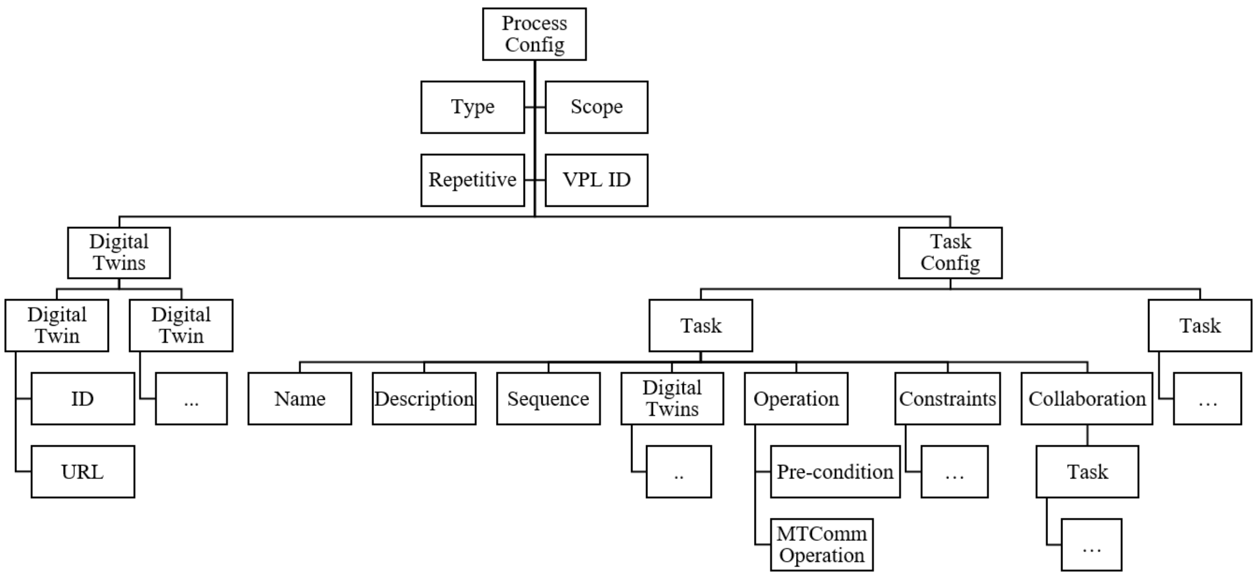


Figure 38: VPL configuration data model for different manufacturing processes. This model refers back to Task model presented in Figure 33 (b).

This workflow outputs a manufacturing process configuration of a VPL. The following describes a configuration data structure to serve the objective of this framework. The figure below (Figure 38) presents a baseline configuration data model based on the requirement of proposed workflow. A process configuration contains metadata describing process type, scope, recurring nature and identifier. It also has a list of cloud-based DTs. List of tasks under a process configuration describes the execution nature of VPLs. Task data structure refers back to Figure 33 (c) and includes some more data items. Sequence number of a task defines when that task will execute in a process. However, in some cases, VPL ignores task sequence number. For example, in an event driven collaborative VPL, task sequence number is not required, hence its value is set for -1. In the case for discrete manufacturing where manufacturing resources can operate independently and in parallel, task sequence is ignored, and its value is set for 0. Therefore, Process Executor component counts task sequence number when it is greater than 0. This process configuration data structure is structured in an XML and stored in VPL Resource Pool component. Each task in process config includes an MTComm's XML based operation message, that is embedded as CDATA so that Process Executor component can send it to DTs in order to execute operation in their respective MRs.

3.5.2.2 *VPL execution workflow*

Here, we present a manufacturing process execution workflow in a VPL using a swimlane diagram in Figure 39. The diagram shows how the Process Scheduler component enqueues task of a VPL based on process configuration. Process Executor component continuously reads from the schedule buffer data list to detect newly requested process execution request. If it is an event-driven collaborative process, Process Executor component enqueues only one task and that task links to the next task until the process finishes. If it is a parallel task, then the Process Executor component enqueues the task and checks with all the DTs of that VPL. It sends the task to the first available DT. This workflow can easily be expanded for other VPL configurations.

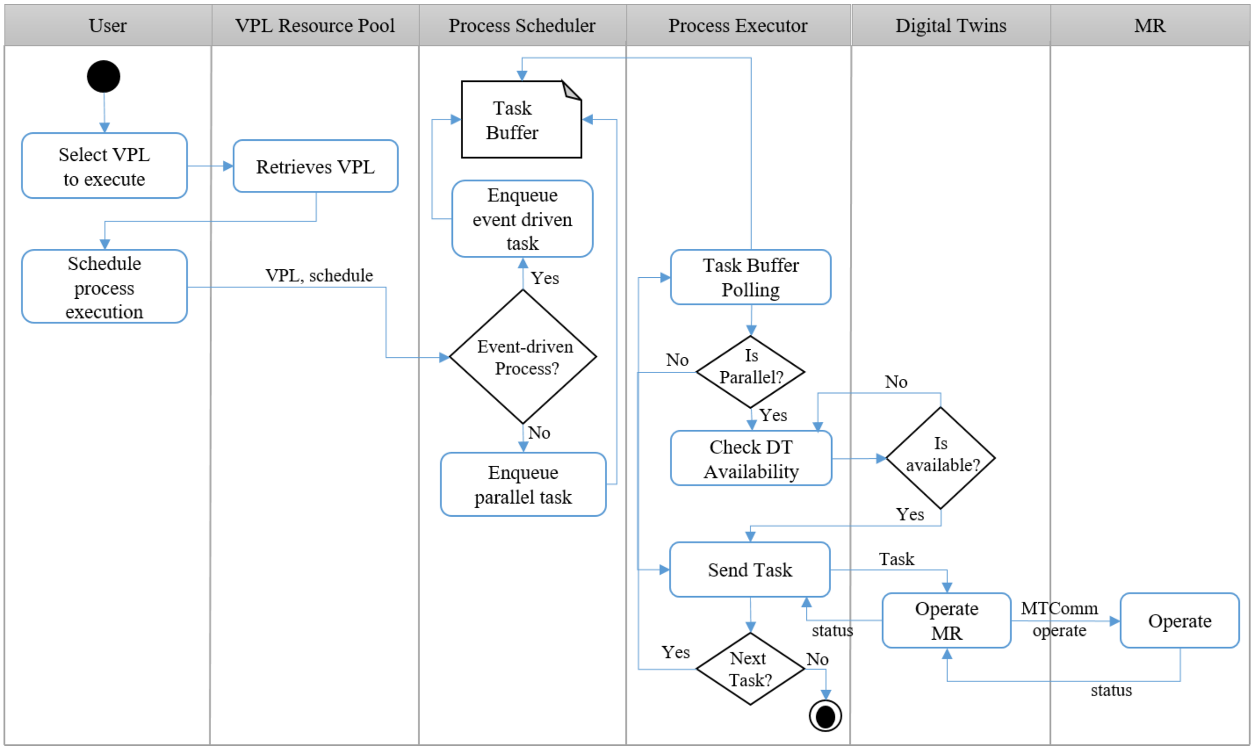


Figure 39: Process execution overflow of DT-based VPL in CPMC where cloud-based DTs operate MR based tasks. This execution workflow supports event-driven collaborative and discrete-parallel manufacturing processes.

3.6 Testbed implementation and case validations

Toward the objectives of this chapter, we detail a prototype testbed implementation and present case validations for proof of concept. First, we describe testbed implementation so that one can reproduce the work. We document implementation approach from the following aspects such as what is the infrastructure; how manufacturing resources are virtualized as DTs using the new proposed MTComm method in cloud; how 3D visualization of machine assemblies will be generated in DTs; how information are organized for DT-to-DT collaborations; and how the proposed VPL configuration workflow is adopted in the testbed. After detailing the prototype, we validate our proposed DT model and VPL framework in CPMC using two case validations for two remote and distributed manufacturing processes.

3.6.1 Testbed infrastructure design

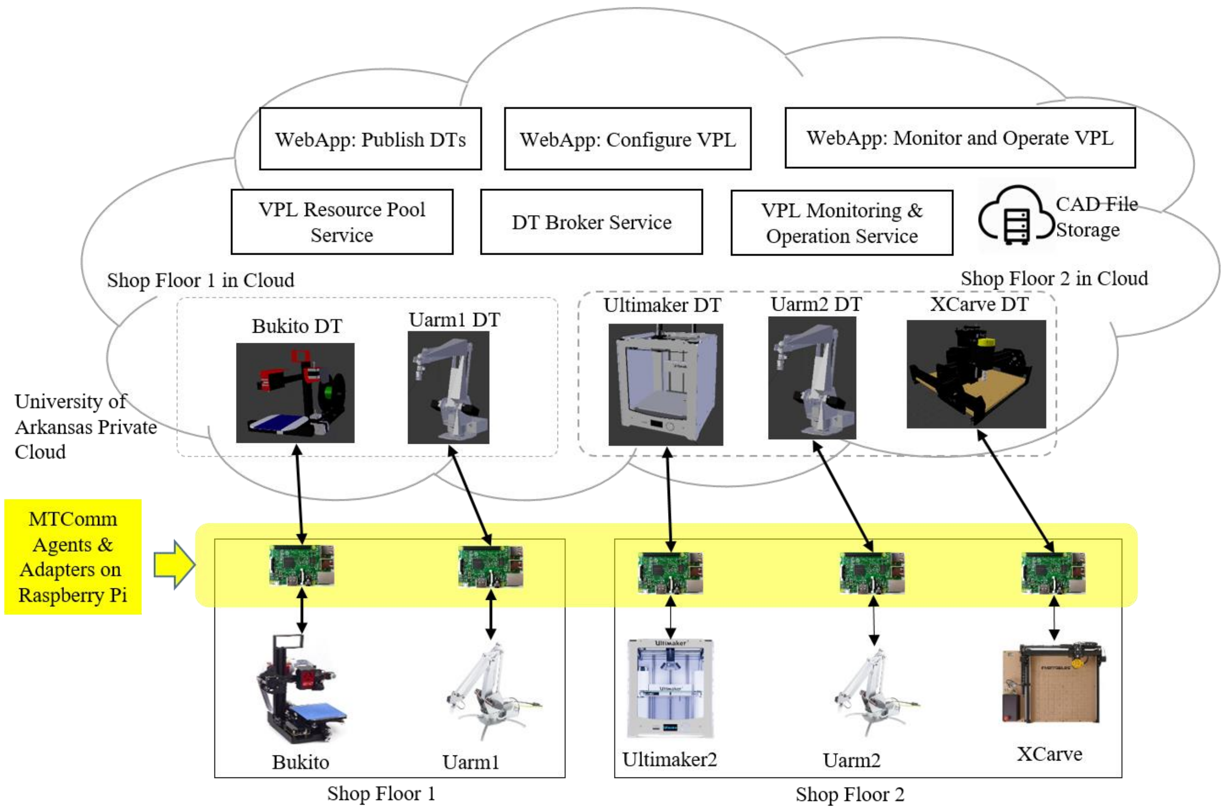


Figure 40: Testbed infrastructure of DT-based VPLs in CPMC that is reconfigurable. There are five machines arranged in two physical shop floors. Each of these machines are virtualized as DTs in the University of Arkansas private cloud. Each DTs are hosted in separate Linux virtual machine instances in the University of Arkansas private cloud. Multiple web applications are developed in a cloud server, which access these DTs to configure, concurrently simulate and remotely operate VPL.

Figure 40 presents the testbed infrastructure that consists of five manufacturing machines located in University of Arkansas (UARK). We arranged the testbed in two shop floors. Shop floor 1 consists of two machines (Bukito and Uarm 1). Shop Floor 2 consists of three manufacturing machines (Ultimaker2, Uarm 2, and X-Carve). The robotic arms are stationary and placed in such an organization that they can collaborate with other machines within shop floor vicinity. Therefore, Uarm 1 can pick up an object from Bukito 3D printer and Uarm 2 can collaborate with Ultimaker 2 and XCarve. Each machine is virtualized as DT in cloud. Each of them is allocated a cloud based virtual machine containers. DTs connect with their physical machines using MTComm method. They download part by part geometries of their assemblies from a CAD File Storage in cloud. 3D assemblies of machines are created using multiple 3D design tools such as Blender, FreeCAD, and Solidworks.

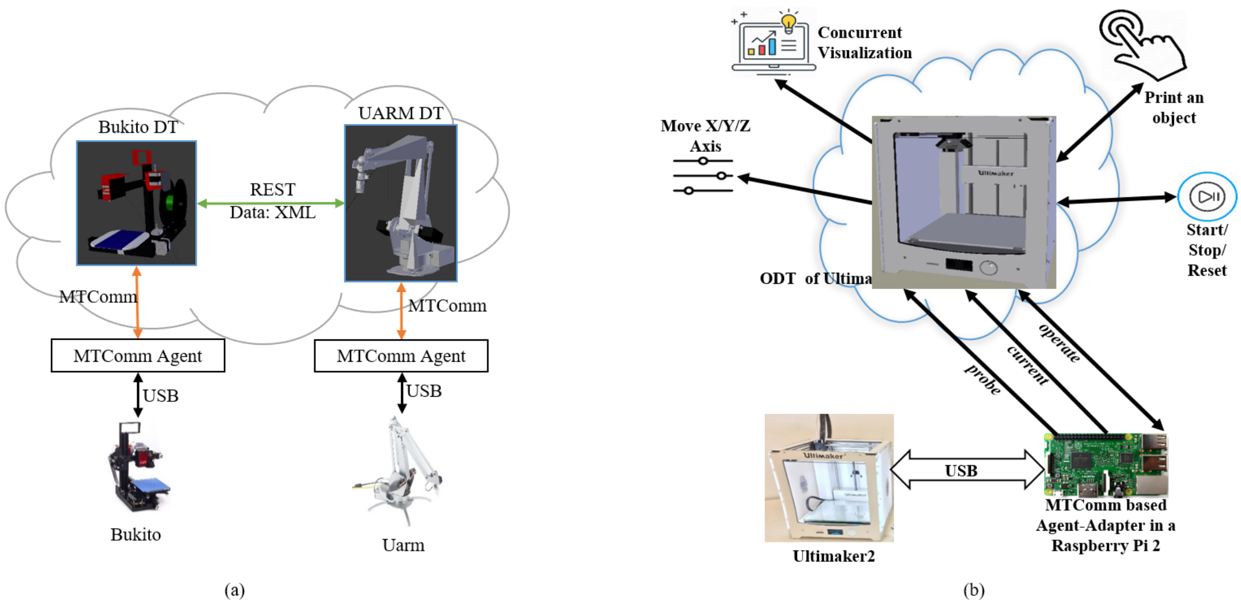


Figure 41: (a) Overall communication methods in the testbed. (b) Virtualizing an Ultimaker2 machine as a DT in CPMC using MTComm.

We also created multiple cloud-based applications and services to make the prototype operational. Following the proposed framework, we created three service-oriented components as follows. DT Broker Service publishes DTs and offers web services for list of suitable DTs while configuring a VPL. VPL Resource Pool component stores the process configuration XML of VPLs after creation. It also retrieves VPL processes on demand. VPL Operation & Monitoring service contains web services for monitoring and executing manufacturing processes. Multiple cloud-based web applications are developed to operate the prototype. We created one application to publish DTs using web services of DT Broker Service component. We developed another web application (Configure VPL) that allows a user to configure VPL of different configurations. Finally, we developed a web application for monitoring and operating a VPL from cloud.

Inside shop floors, each machine tool is connected to a Raspberry Pi device that works as a MTComm Agent-Adapter based middleware devices between the machine and cloud. A token-based authentication is implemented to secure the manufacturing resources from unauthorized access. We present a communication map of the implemented testbed in Figure 41 (a). It shows communication methods for different aspects of the testbed.

3.6.1.1 *Implementing virtualization of physical machines as DTs using MTComm*

Figure 41 (b) presents the virtualization of Ultimaker2 physical device into its DT using MTComm. Ultimaker2 DT uses *probe* web service to collect semantic ontology information of the physical device. Ontology Parser subcomponent from the component architecture of DT in Figure 29 extracts MTComm based virtualization information. Based on virtualization information, Ultimaker2 DT collects near real-time status data of physical machine using the *current* web service. It executes physical operations through the *operate* web service.

Figure 42 (a) presents a state diagram that shows the programming flow in parsing MTComm *probe* XML response document in the OP subcomponent of DTs in Cyber-Physical Manufacturing Cloud. This state diagram recurrently visits all the leaf nodes of the XML tree and takes decisions to store the virtualization information in the Virtualization Database. If the discovered information is a Data Item for monitoring, OP persists the information as a monitoring information. When OP discovers an Operation element in the XML tree, the it persists the information as operation information in the Virtualization Database.

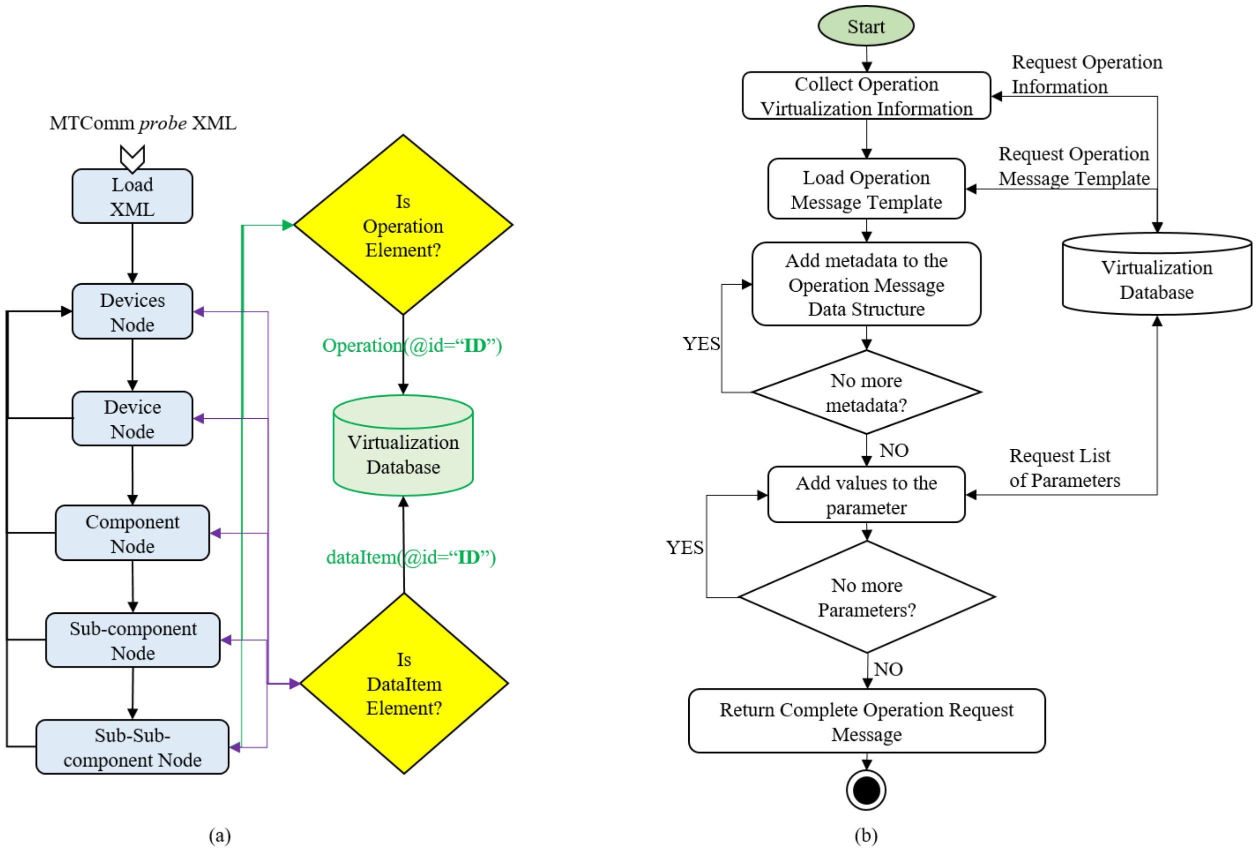


Figure 42: (a) A state diagram to describe extraction of virtualization information from MTComm probe document in a DT. (b) A flow chart to generate MTComm's Operation Request message.

When sending an operation request through the *operation* web service, the DT constructs an XML as body of the HTTP POST request. Construction of the operation request message is presented in Figure 42 (b). It starts with collecting virtualized physical operation from the Virtualization Database. It loads a template XML from the same component to start crafting the XML message. On the path towards constructing the message, it adds required metadata and parameters of the virtualized operation. The program ends by returning the constructed XML message, which executes an operation in the physical machine.

3.6.1.2 *Constructing 3D Visualization of implemented DTs in testbed*

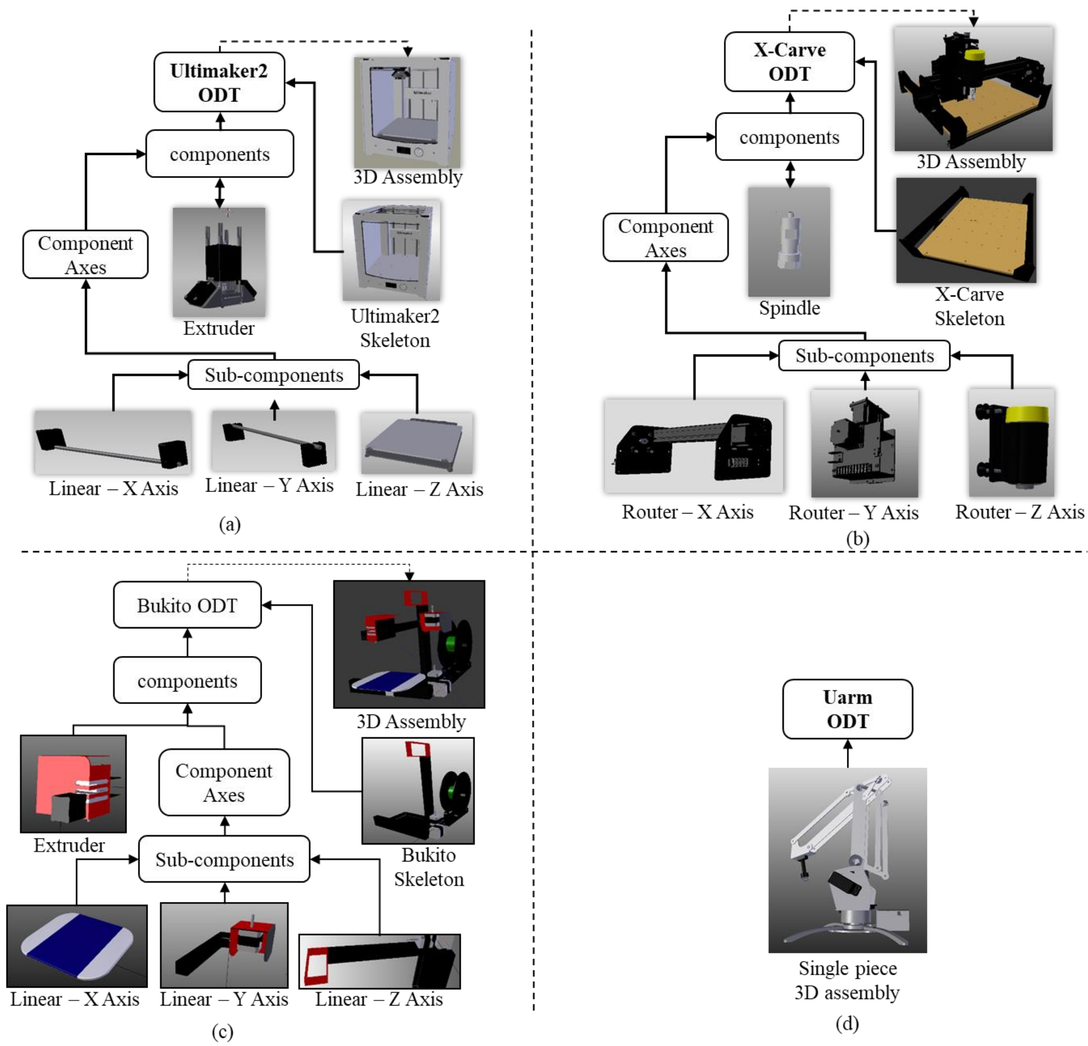


Figure 43: Bottom-up aggregation of 3D visualization in DTs from the testbed. (a), (b), (c), and (d) presents aggregation of Ultimaker2, X-Carve CNC, Bukito, and Uarm respectively.

Proposed DTs create high-fidelity 3D simulation by aggregating geometries of machine parts. A cloud-based application can access 3D visualizations of DTs via web services. 3D graphical visualization of the DTs is rendered in a bottom-up aggregation approach. We developed an aggregator computer program using JavaScript based Three.js framework that downloads part 3D assemblies of DTs from the CAD Model Storage. Downloaded part assemblies are then aggregated from granular level following the semantic ontology described in the probe response XML of MTComm. Part by part aggregation of 3D assemblies of four different machines from the implemented testbed are presented in Figure 43. Each level (device or component) in the ontology comes with the URL of their downloadable 3D assemblies as previously shown in the MTComm ontology model. The exception was in case of Uarm DT, where a single piece 3D assembly was developed during construction. Movements of robotic arms are much more complex than the 3D printers and CNC machines and required rules of physics applied in the geometry. We use Blender armature to orchestrate animations in the 3D assembly of Uarm robot.

3.6.1.3 *Collaboration information of DTs*

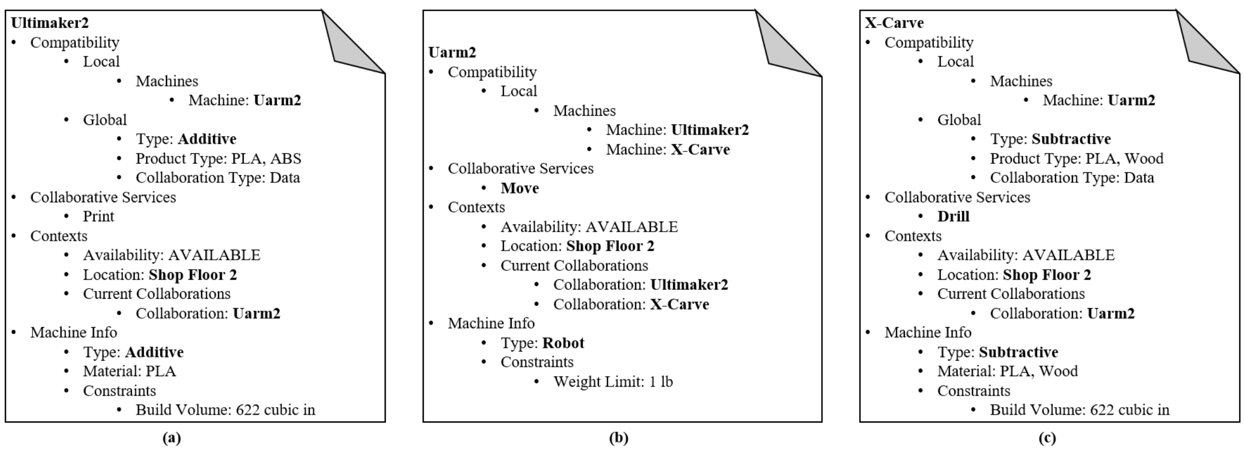


Figure 44: Collaboration information for all five implemented DTs. This information is applied while configuring a VPL.

Figure 44 presents collaboration information of three collaborative DTs from the testbed. This information is structured following the model presented in Figure 33 (c). This information is utilized for validating collaboration when a VPL is configured by WebApp Configure VPL in testbed. We describe more on an event-driven collaborative scenario in next section. The figure above shows information presented in three XMLs and their differences are typed on bold fonts. The figure shows Uarm 2 can collaborate with two other (Ultimaker2 and X-Carve), while Ultimaker2 and X-Carve can collaborate with Uarm 2 but not with each other.

3.6.2 Case validations

3.6.2.1 *Case validation 1: Configuring a VPL consisting of MRs across shop floors*

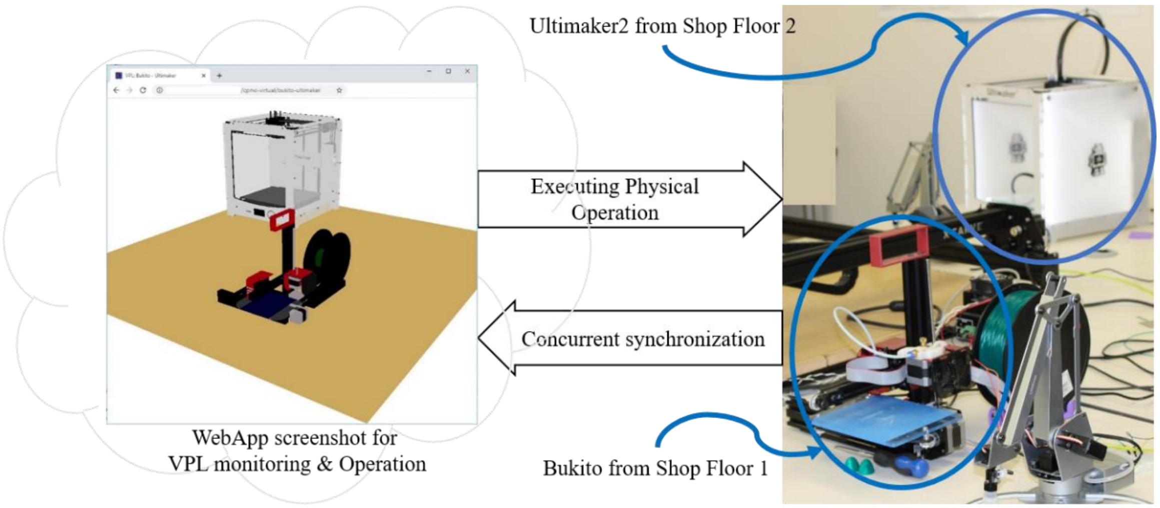


Figure 45: A DT-based VPL in CPMC consisting of two DTs from two shop floors.

Use case statement: *As a virtual manufacturer, I want to configure and administer a distributed virtual production line consisting of discrete 3D printer machines, so that I can produce 3D printed artifacts whenever I want.*

Figure 45 presents a configured VPL that can satisfy the above use case statement. This VPL is configured following the workflow in Figure 37. It consists of the Bukito DT from Shop Floor 1 and the Ultimaker2 DT from Shop Floor 2. It establishes a discrete manufacturing process where both the DTs can operate in parallel. A screenshot from web application Monitor & Operate VPL (from Figure 40) and a picture of the physical testbed are taken to demonstrate the use case infrastructure. Execution of manufacturing process in this VPL follows the workflow presented in Figure 39.

3.6.2.2 *Case validation 2: Reconfiguring a VPL within a shop floor*

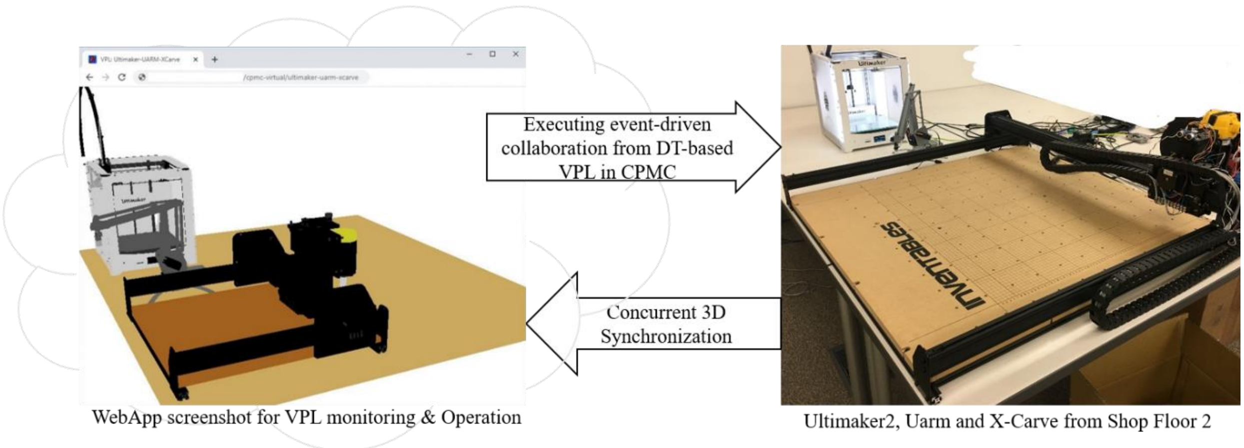


Figure 46: A VPL consisting of three DTs and they are all from Shop Floor 2.

Use case statement: *As a remote manufacturing operator, I want to reconfigure manufacturing resources in my physical shop floor to collaborate in an event-driven production process, so that I can adjust my production lines for new orders.*

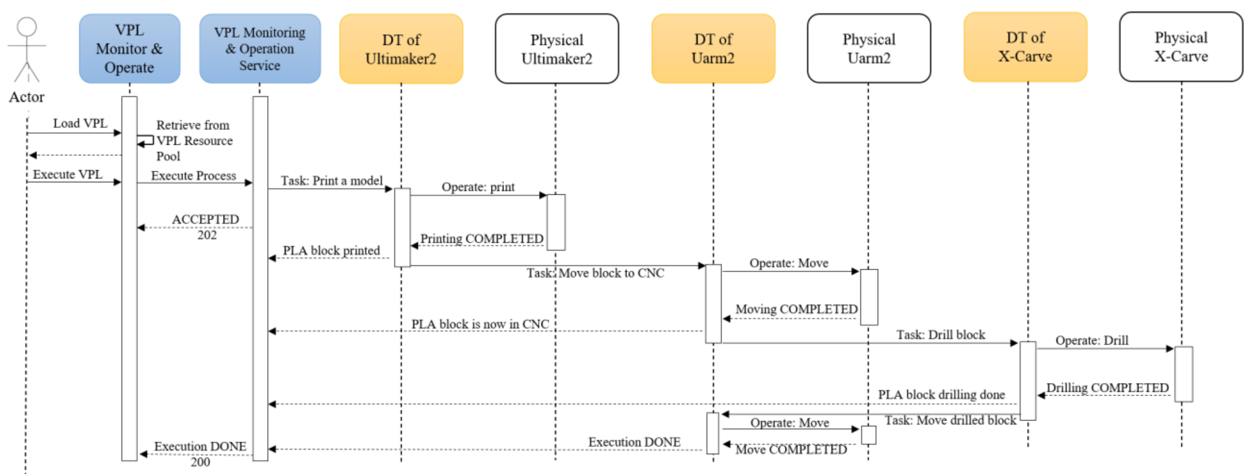


Figure 47:A sequence diagram to describe DT-to-DT virtual collaborations in CASE VALIDATION 2.

Figure 46 presents a mapping between DTs in the configured VPL and their respective physical machines in the testbed. This VPL consists of three DTs of three following machines such as Ultimaker2, Uarm2 and X-carve. These three machines configured to collaborate in an event-driven manufacturing process. Manufacturing process execution of this workflow follows the same path as shown in Figure 39.

This VPL is configured following the workflow of Figure 37. While configuring this event-driven process we gave input for four following tasks such as i) print a 3D model file (assigned to Ultimaker2 DT); ii) move the printed object from 3D printer to the CNC machine (assigned to UARM2 DT); iii) drill on top of the printed and moved object (assigned to X-Carve CNC DT); iv) move the object from the CNC machine (assigned to Uarm2 DT). The task is configured according to the model of Figure 33 (b) and each task triggers the next task on event ‘COMPLETED’. Figure 47 presents a sequence diagram to demonstrate event driven collaborations among DTs of Ultimaker2, Uarm robot, and X-Carve CNC machines.

3.6.3 Discussion

This research presents a concept of configuring VPL using DTs in CPMC. In this section, we contrast some fundamental features of this chapter with comparable state-of-the-art. It is also worth addressing the limitations and challenges of this study.

3.6.3.1 *Feature wise comparison with state-of-the-art*

Here, we compare our proposed framework with four comparable state-of-the-arts in Table 1. Reiterating objective of this work, we propose a framework to configure, monitor and operate DT-based VPLs in CPMC, so that a remote manufacturer can orchestrate a virtual manufacturing process in cloud using DTs instead of individual web services. Features of comparisons are chosen to contrast the conceptual differences of our work with these existing researches. We choose the following five features of our framework for comparison. First, we compare on the basis of manufacturing paradigms such as cyber physical production systems (CPPS), cloud manufacturing

(CMfg) and CPMC. Second, we draw comparison on the basis of whether these works propose cloud-based DT architecture. Third comparison is done with the question of whether they describe a reconfigurable VPL framework. Fourth feature to compare is capability of configuring VPL from cloud. Final fundamental feature is capability to configure distributed VPL across shop floors.

Table 1: Feature wise comparison between this work and other state-of-the-arts to contrast the key conceptual differences.

State-of-the-art → Features ↓	Ding et al. [58]	Liu, Jiang, & Jiang [69]	Park et al. [68]	Yu & Xu [67]	This Research
Manufacturing Paradigm	CPPS	CPPS	CPPS	CMfg	CPMC
Cloud-based DT Architecture	No	No	No	Yes	Yes
Reconfigurable VPL Framework	Yes	Yes	Yes	No	Yes
Configuring VPLs from Cloud	No	No	No	No	Yes
Distributed VPL across shop floors	No	No	No	No	Yes

This table of comparison shows that there is a significant research gap in cyber physical system enabled manufacturing clouds where manufacturers can configure production lines remotely. It is suggested in [62] that modern manufacturing systems are moving towards a direction where manufacturing resources no longer become operational for shop floor owners. Rather there will be users outside the shop floors. This is also one of the motivations towards building manufacturing cloud concepts and architectures. Our research fits to manufacturing solutions where remote manufacturers themselves want to monitor, and control shop floor machines and configure production lines remotely. This research is a proof of concept and proposes initial design. This table of comparison gives an insight towards future research directions.

3.6.3.2 *Limitations and challenges*

Concept of DTs evolved around machining data acquisition and predictive analysis in product lifecycle management. In our research, we explain a technique for data acquisition. However, we have not incorporated any predictive analytics in this work. Predictive analytics can be done in two layers of the proposed framework. Collected data from manufacturing resources remain within the DT Management Layer where fault diagnosis and predictive fault analysis can be done in future research. Collected data for production processes of VPLs is stored in VPL Management Layer, where predictive analysis of manufacturing processes can be done.

Another limitation in this work is lack of self-adaptiveness in response to a machine fault. DTs are gaining rapid popularity in autonomous systems where they can take decisions based on fault diagnosis and predictions. Our proposed DT architecture is expandable due to its service-oriented architecture. Objective in this chapter is to design an architecture and framework, and present case studies to prove feasibility of concepts. Future work should include self-adaptive VPL for manufacturing clouds.

Manufacturing configurations are fundamental to VPL structure and dynamics. There are several manufacturing configurations investigated in [67, 62]. There are numerous other articles that propose different manufacturing cloud architectures for different manufacturing configurations. Hence, it is of critical need to survey the existing service oriented VPLs in manufacturing clouds. A future work should address the advantages and performance improvements using our proposed concept.

During the implementation of DTs, we faced three major challenges in concurrent simulations. Firstly, we had to design CAD based 3D models of the machines part by part. Each part had to be observed and their individual 3D assemblies had to be created. There are others popular ways to generate 3D model of physical objects including, but not limited to 3D scanning, VR, and AR. However, these methods are not suitable to apply for rendering 3D assemblies with a detailed granularity. Secondly, due to loading heavy 3D models of the physical device in cloud-based applications, there were significant delays in near real-time simulation of machine movements. [65] presents network delay analysis in rendering 3D visualizations of DTs. In our experiment with the testbed, we observed that the delay increases proportionally with loaded 3D models in the cloud application for near real-time 3D simulation. Thirdly, DT renders a high-fidelity 3D assembly of the corresponding physical device in cloud. In our implementation, we developed an application where the 3D assembly is rendered, and a simple GUI is generated for the user to click and execute physical device operations. However, this GUI can be improved by shifting the interactions within the 3D model of the physical device. A better implementation of 3D visual and interactive DT would be to render it in a VR environment, where a human operator can join the VR scene using headsets. Therefore, human-DT interaction can be potential future topic of research. Last but not the least, future work should reinvestigate the process of creating a DT using cloud-based containers. It is desirable to design a new cloud-based container model specially tailored for DTs of manufacturing resources.

3.7 Conclusion

Emergence of Cloud Manufacturing paradigm has enabled manufacturers to configure Virtual Production Lines (VPL) remotely where manufacturing resources can remain in geographically distributed shop floors. Recently introduced Cyber Physical Manufacturing Clouds (CPMC) virtualizes physical manufacturing resources as web services. CPMC allows a user to directly monitor and operate physical machine tools from cloud via the virtualization web services. In such contexts like CPMC, a user can subscribe manufacturing web services to configure a VPL.

But subscribing a web service of a virtualized manufacturing resource allow users to access only one or partial functionalities, the user has to subscribe many services to gain full access. This issue leads to difficulty in manufacturing resources monitoring and operations. It also impedes reconfigurability of VPLs when a change occurs. In this chapter, we investigate a new virtualization approach using cloud-based Digital Twins to configure, monitor and operate VPL instead of web services.

We present a service-oriented architecture of DT in CPMC where each manufacturing resources are virtualized as DTs. The architecture design enables a user to monitor and operate physical manufacturing resources directly from cloud. We describe a virtualization method based on ontology model of MTComm to transform manufacturing resources as DTs. This DT architecture also facilitates a technical structure that can configure DT-to-DT collaborations from cloud. We conceptualize VPL in CPMC based on the proposed DTs and different production process configurations. To support the concept, we present a framework to configure, monitor, and operate DT-based VPL in CPMC. We detail the framework components, configuration workflow, and execution workflow. A testbed is implemented to evaluate the proposed architecture, methods and framework. We developed few web applications in cloud to publish DTs and configure them into multiple VPLs. Implementation details are documented in this chapter to make this work reproducible. We use two use cases to validate the proposed framework and discuss the results. Results from the experiments show DT-based VPL offers high reconfigurability for different manufacturing configurations in CPMC.

CHAPTER 4: A FRAMEWORK FOR PUBLICATION AND DISCOVERY OF DIGITAL TWINS

***Abstract* -** Traditional service registries or catalogs publish and describe individual services of different entities. In this chapter, we propose a new approach of service publication and discovery based on the concept product catalogs. In this new approach, entities are equivalent to products in traditional products catalogs. We conceptualize an entity registry where each entry constitutes to its collection of remote services. To abstract entities, we utilize the concept of Digital Twin (DT). To support our objective, we present a reference DT architecture that virtualizes entities, exposes all its functionalities as services, and offers a remote programmable instance to invoke the services directly from application code. To publish and discover DTs, we propose a novel framework, OpenDT. This framework enables entity owners to publish DTs and allow others to discover them for different purposes such as creating mashups and applications using the DT services. It also allows developers to create composite DTs consisting of other DTs for large and complex entities. To evaluate OpenDT, we implement a cyber-manufacturing testbed comprising of multiple machine tools and their DTs. Case validations show excellent efficiency in entity publication and discovery using proposed DTs.

4.1 Introduction

Traditional service registries or catalogs act as central sources of digital services where organizations publish individual functionalities of their entities as remotely accessible services and manage them in way so that users can search and discover them [87, 88]. However, publishing each service separately leads to overpopulation in service registries, which can potentially impede service discovery performance [89]. Hence, it is desirable to design a more integral approach in service publication. In this chapter, we investigate a new approach akin to the concept of "product catalogs", where entities are equivalent to products and published as a collection of services compared to the existing individualistic publication approaches.

To contextualize the problem statement, we consider the case of a popular service publication platform called ProgrammableWeb [90], where different organizations can publish their services and application programming interfaces (API) to enable public access to their entities. This platform stores and indexes each of the services and APIs in service registry with different search tags. For example, search key 'Mercedes-Benz' returned a list of 11 different APIs, which consist of many web services for different functionalities such as remote vehicular diagnostics support, fuel status monitoring, automotive configurations etc. To create a comprehensive monitoring and diagnostics application for a specific car model, one has to discover and subscribe web services across these APIs. Services required for a specific car model are scattered across all these APIs, which may lead to inefficient discovery during runtime and offline. Additionally, a third-party developer can find it difficult to determine car model to API compatibilities in this existing service registry.

In this chapter, we propose a new service publication approach where we publish virtual representations of various type of entities. Each of these virtual representations combines all essential services of an entity into one package. Thus, one can subscribe all services of an entity from a single entry of service registry. To design such virtual representations of entities, we address three fundamental requirements. First, they need to suitably abstract corresponding entities in a virtual space. Second, they must have service-oriented architectures to accommodate different elements of service computing. Third, they must offer easy programmability so that programmers can access them from their application code.

Considering these requirements, we identify Digital Twins (DT) as a primary candidate in our proposed approach to create virtual representations of entities. DT is a rapidly growing concept in the domain of smart manufacturing, which creates virtual copies of entities and offer entity functionalities as services [56]. We present a new service-oriented architecture to satisfy the three above requirements. Presented architecture creates virtual representations of corresponding entities. It documents all remote services of an entity in a service descriptor to offer interoperability. It also offers remote access to entities by accessing their virtual instances. There is another motivation towards using DT as the primary tool in our proposed service application approach. Particularly, when published services represent dynamic functionalities and behaviors of Internet of Things (IoT) and Cyber-Physical System (CPS) enabled large physical entities [91, 89]. Because, DTs can create synchronized instances of such entities in virtual space [92, 65].

Supporting the objective of this research, we propose a novel service-oriented DT publication and discovery framework, namely OpenDT. It allows entity owners to create proposed DTs and publish them in OpenDT. Users can subscribe and use published DTs in their mashups and applications. One can also create a new composite DT by combining multiple published DTs.

The followings are our research contributions.

(i) We develop a new service publication approach based on the concept of "product catalogs". In this new approach, collection of services of entities are published instead of individual services.

(ii) We choose DT as the primary tool to package collection of services of entities and represent them in virtual space. To underpin this choice, we present a new service-oriented architecture of DT that offers easy remote programmability.

(iii) We design of an open framework, OpenDT, to publish and discover DTs of various entities.

Rest of the chapter is organized by the following sections. Section II presents a motivating scenario and addresses different research challenges. Section III presents the general approach, OpenDT framework, SoA of DT, and other essential elements of the framework. Section IV presents an implementation of fully operational OpenDT framework and presents case validations. It also presents related works and draws state-of-the-art comparisons. Section V concludes the research and indicates future work.

4.2 Motivation

In this section, we present a motivating scenario towards designing the new approach. We discuss the necessity of a new service registry approach based on the motivating scenario. We also address the research challenges to solve the conceptualized problem of the motivating scenario.

4.2.1 Motivating Scenario

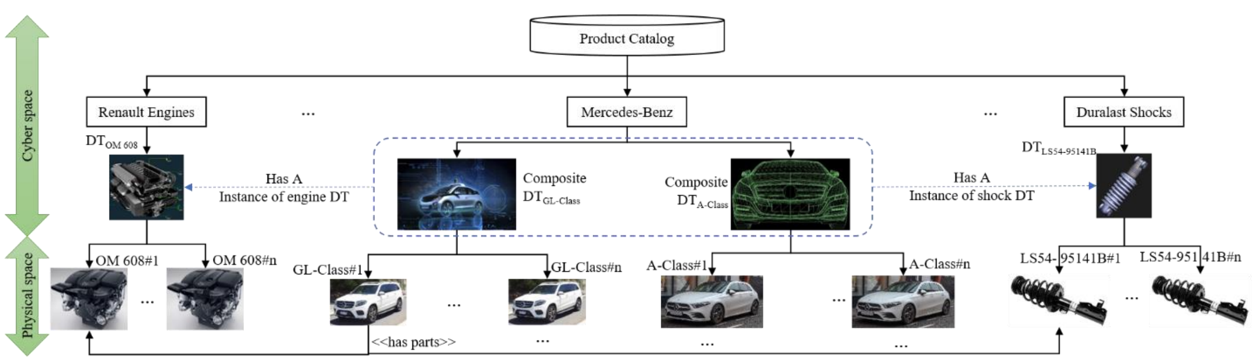


Figure 48: A motivating scenario to describe the new publication approach based on the idea of product catalogs.

Figure 48 depicts a motivating scenario to conceptualize the new publication approach. The following describes it.

Auto manufacturer Mercedes-Benz manufactures different models of cars. Each car comes with common components such as engine, brakes, absorber shocks, struts etc. Mercedes-Benz does not manufacture all of these parts. Some of these they from different other manufacturers. For example, Renault S.A. produces engines (model OM 608) for different models of Mercedes Benz car models such as A-Class, GL-Class, and others. Similarly, automotive manufacturer Duralast produces shocks (model LS54-95141B) for these cars. These automotive parts are equipped with IoT sensors and their status data can be collected by readings from Controller Area Network (CAN) busses of cars. We assume for this case that these sensor data readings are limited to the Mercedes-Benz CAN busses because they come from different organizations. Mercedes-Benz publishes several APIs in a service registry for collecting vehicular data, so that third party application developers can develop mashups for different purposes such as monitoring, diagnosis, sales, etc. Renault and Duralast publishes APIs for their products, which offer more information than the CAN busses of cars.

We now live in an era where each business are entities are going through digital transformations. Web services have been instrumental in digitalizing different types of entities. Emergence of IoT and CPS excelled the digitalization, which also enabled service-oriented monitoring and control of physical things. In addition, digitalization also creating a technical universe of interconnected things or entities. Motivating scenario of Figure 48 is one of the numerous interconnected automotive entities, where a product (GL-Class) consists of many other sub-products (OM 608 & LS54-95141B) from different organizations. To create a remote monitoring and diagnosis mashup application for the cars, developers need to subscribe remote services from Mercedes-Benz, Renault, Duralast and others. In traditional UDDI-based service registries, these remote services are scattered all over the data repositories. The following sections investigate how these scattered services can be organized efficiently, so that a developer can access subscribe and access them from a single source.

4.2.2 Need for a novel product-oriented publication approach

We begin by discussing two issues associated with the remote monitoring and diagnosis application stated in the motivating scenario, if it is developed using traditional UDDI-based service publication approaches.

First, we point out that there is a mismatch between involved real entities and their logical representations. If a car (GL-Class) in Figure 48 is observed from object-oriented design (OOD) approach, we can see that it consists of many other parts such as Renault OM 608, Duralast LS54-95141B, etc. Therefore, an OOD of GL-Class model should consist of instances of those part models. Such an OOD is absent if the application is developed using traditional service registries.

Second, we identify an issue in runtime discovery of remote services. When a diagnosis application measures car health, it needs to access remote services of different parts, which may need to come from Original Equipment Manufacturers (OEM). Hence, the application needs to discover these OEM provided services, which warrants additional remote service calls for discovery and can impede application performance.

Addressing the above issues, we envision a new publication approach as depicted in Figure 48. Our end product of this research is to develop a product catalog where organizations will register their product models as multi-tenant DTs. These DTs will have service-oriented architecture and can be composited based on their ontology structure. Thus, Renault will create a DT for all engines of model OM 608, denoted by DT_{OM608}. Similarly, Duralast will create a $DT_{LS54-95141B}$. Mercedes Benz will create two DTs namely DT_A and DT_{GL} for all car models of A and GL respectively. In addition, both DT_A and DT_{GL} will have personalized instances of DT_{OM608} and $DT_{LS54-95141B}$ matching their physical ontology. All of these DTs will have a descriptor that documents each of their remote services. Last but not the least, these DTs will enable anyone to access them from code so that one can create a new DT for a new product. Thus, other car manufacturers can subscribe and access DT_{OM608} and $DT_{LS54-95141B}$ in order to create new composite DTs for their car models.

4.2.3 Research Challenges

We address the following four research challenges towards the research objectives.

Logical representation of entities: There are many ways an entity can be abstracted in a virtual space. One of the ways is using REST protocol, which takes an object-oriented approach to transform entities as logical resources. However, stateless nature of RESTful representation may not be applicable in some cases of CPS and IoT enabled physical entities [88, 89, 91], where our motivating scenario lies. Hence, it is desirable to investigate another approach of virtually representing entities. From an empirical analysis from the domain of smart manufacturing, we identify the concept of DT as a potential candidate to logically represent different types of entities.

Service-oriented architecture of DTs: Existing architectures of DTs are designed to satisfy the requirements of smart manufacturing [92]. Therefore, state-of-the-art DTs have been utilized to virtually represent machine tools for different purposes such as remote monitoring, visualization, control, predictive analytics, etc. In the context of this research, we need to investigate a more general service-oriented architecture of DTs, so that it can represent entities beyond the domains of CPS and IoT.

Service discovery using DTs: Traditional UDDI-based service repositories [88, 89] store individual remote services, which creates numerous data records in registry databases. To reduce the number of scattered remote services, we investigate how number of registries can be reduce by publishing entities as DTs. We also investigate how a DT can help mashups and applications to reduce latency in service discovery.

Remote programmability of DTs: Another research challenge towards the goal of this research is to design DTs as remotely programmable instances so that one can easily combine them to create a composite entity. Existing service registries and catalogs only publish, describe and manage services. However, many of them do not offer a remote programmable instance to invoke the services. For example, in ProgrammableWeb, users can only publish APIs but cannot import and access programmable objects to call them. We investigate a method where programmers can import objects of entities, which expose functionalities as services.

4.3 Reference framework of OpenDT

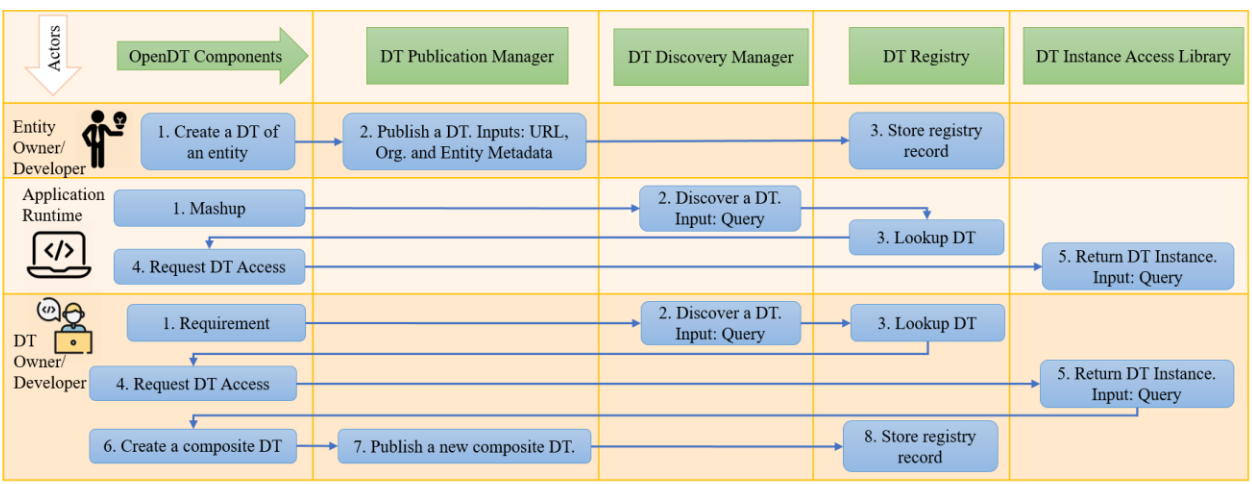


Figure 49: A high-level view on OpenDT framework. This diagram presents basic operations of OpenDT such as DT publication, registry, and discovery.

Towards the research goals presented earlier, we design a framework, namely OpenDT, which establishes a publication and discovery platform based on the idea of *product catalogs*. Figure 49 depicts an overall picture of the framework by showing interactions among different actors and components. Rest of the section organizes as follows. First, the general approach of the OpenDT framework is detailed. To support the framework, a service-oriented DT architecture is discussed and formalized. This section also discusses a DT descriptor model that provides the foundation of service bundles within a DT. Last but not the least, a query architecture for DT service discovery is presented.

4.3.1 Framework design

OpenDT diagram in Figure 49 depicts a general approach and an overall workflow for publication, discovery and remote programmability of service-oriented DTs. Followings discuss the approach by describing different actors, components and their basic operations in the framework.

4.3.1.1 *Actors*

We consider three types of actors in the framework. Entity Owners create DTs for their entities and publish them using different components of the framework. Published DTs are discoverable by other users for use. Third party developers can discover and access DT services from code and create applications and mashups for general users. During runtime, application code can dynamically discover and access the published DTs. Some developers, who belong to different organizations, can discover DTs that their products have as parts and composite them to create new composite DTs.

4.3.1.2 *Components*

DT Publication Manager (DTPM) component offers services where entity owners can publish their DTs. There are existing service publication platforms, such as ProgrammableWeb. This DTPM module functions similar to that platform. However, the fundamental difference is DTPM publishes DTs instead of web services and APIs.

DT Registry (DTR) component stores registry information produced by DTPM component. This component contains a database for the registry entries. It also keeps necessary descriptive and metadata of DTs.

DT Discovery Manager (DTDM) component allows users to search for suitable DTs and their services. When searching DTs in runtime, this module discovers list of suitable DTs and shows them to the requesting mashup or application.

DT Instance Access Library (DTIAL) component offers a plugin that is a pre-compiled library for mashup and application developers. This precompiled library channels a communication between the OpenDT framework and applications. Upon request for a specific DT, this plugin component retrieves the reusable DT instance, which is programmable in the application code.

4.3.1.3 *Basic operations*

Among many basic operations of service computing, we consider three, which are fundamental to the research contributions of the reference framework. They also help us describe the general approach of OpenDT.

DT Publication: In order to publish, entity owners submit the endpoint of a DT with descriptive information and metadata. Upon submitting, DTPM component probes the DTs and automatically generate entry for the DTR component. Finally, DTR stores the publication information into its data store.

Runtime discovery of DTs: Cases of runtime discoveries occur from mashup or application codebase. A mashup application configures a query and send it to the DTDM component. DTDM analyzes the query information and preprocess it for DTR to discover the suitable DTs. DTR component returns URLs of the DTs so that they can be requested for access to DTIAL component.

Compositing a new DT: In previous sections, we described the approach of creating composite DTs of entities, which are consisted of several other DTs of sub-entities. Hence, we describe a unique composition operation in OpenDT. Developers from different organizations can discover the sub-entities, which construct their products and get necessary DT instances upon request to DTIAL. These DT instances are programmed into a new composite DTs for new entities and published into DTR. This way OpenDT establishes a network of interconnected DTs.

4.3.2 Service-oriented architecture of DTs

There are some properties of a DT, which are fundamental reason behind the design of OpenDT. In addition to the requirements, we consider two of the DT properties while designing the service-oriented architecture in Figure 50. First, there are a one-to-many relationships between DTs and their virtual representations. A study in [93] shows four different information models of a single additive manufacturing machine.

We also consider homogeneity as another key property of DTs, which is particularly useful to represent heterogeneous entities. A DT of an entity can be re-instantiated and reused in other contexts or organizations. Inversely, a DT can represent and work with many entities of a type. Homogeneity property of DTs helps the developers to create them in a uniform method [92].

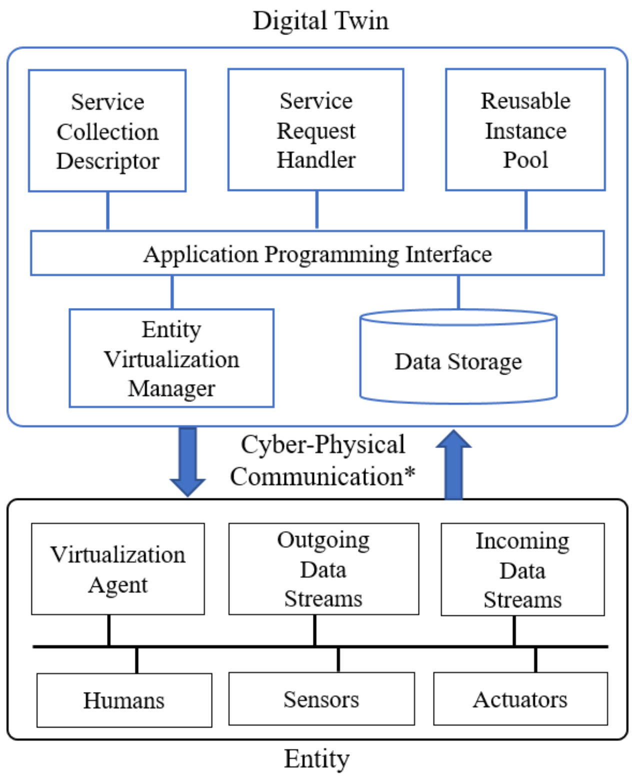


Figure 50: A service-oriented architecture of DT. Cyber-Physical Communication is applicable in the contexts of CPS and IoT.

We divide the diagram into two parts such as the entity, and its representational DT in virtual space. An entity can be entirely logical where the corresponding DT performs simulation of it. On the other hand, an entity can be physical, particularly in the contexts of CPS and IoT enabled entities, where they integrate computational and networking capabilities and establish cyber-physical concurrent synchronizations. In such contexts, they exchange real-time data back and forth with the DT [65]. Data exchange from entity to DT corresponds to real-time monitoring data collection of physical processes. Inversely, DT to entity data exchange resembles remote service-based execution of entity functionalities. The following sections formalizes a general concept of entities and their corresponding DTs.

We identify different aspects of the proposed architecture and develop Moore's [21] finite state machine. The proposed architectural system, denoted by S_{DT}, consists of the following subsystems and components such as set of entities EN, their DTs DT, Entity Virtualization Manager V, Service Collection Descriptor Svc, Reusable Instance Pool Lib. The following expression formalizes the architectural system.

$$S_{DT} = \{EN, DT, V, Svc, Lib\} \qquad (1)$$

4.3.2.1 *Entity*

Physical components of an entity comprise of entities consist of one or multiple humans or devices. The devices have Sensors and Actuators. All of these elements are virtualized using the Virtualization Agent component. For simulation-based DTs, input data can be collected offline, while for DTs with the property of Multiphysics, two components take the data transmission responsibilities. Outgoing Data Streams component gathers status information from the physical components and transmit to DT. Inversely, Incoming Data Streams [66] component accepts and routes requests for entity functionality execution.

Here, a formal presentation of CPS and IoT enabled entities is constructed. We adopt the CPS based design of physical objects in our design [94]. We consider that one entity can consist of one or more smaller entities. An entity, denoted by en $\in EN$, is comprised of the following eight elements such as sensors S_e, actuators A_e, Incoming Data Streams C_e, set of events E_e, Outgoing Data Streams D_e, and virtualization agent V_e. The following expression formalizes

$$en = \{S_e, A_e, C_e, E_e, D_e, V_e\} \qquad (2)$$

Considering the formation using things, we formalize entity by the following equation.

$$EN = \{en_i, where\ i = 1 \ldots |EN|\} \qquad (3)$$

4.3.2.2 *DT*

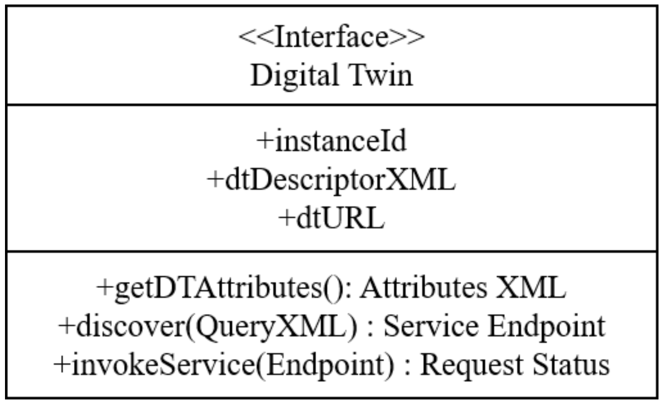


Figure 51: Class diagram of Digital Twin interface. This interface includes general attributes and methods of a DT. DT developers define implementation of this interface for use of applications and mashups.

Now we discuss the components within DTs. Entity Virtualization Manager component defines the business logics, endpoints of functionalities and semantic structures of virtualized entities. The entire virtual representation of entity relies on this component. It organizes and stores data in the Data Storage component. We consider the Data Storage component as a single multi-tenant database, which facilitates DT multitenancy. For simplicity of design, we assume that it is scalable like the traditional big data storages. This component is instrumental towards forming the different information models of DTs such as Knowledge-based Models, Graph-based Models, Mathematical Models [93], Geometry-based models [65], Offline Simulation-based Models [56].

To construct a DT, entity owner needs to publish its functionalities as services, which can be done using traditional approach of APIs. Thus, we put API as a component within the DT part. API uses Entity Virtualization Manager and Data Storage components to virtualize entity functionalities and characteristics as services. Service Collection Descriptor module stores an

UDDI-based document that describes the collection of services hosted in the API component. We discuss more about this component in next section.

Service Request Handler component receives incoming service requests from applications and mashups, validates and forwards them to the API component. While routing incoming requests, it accepts dot notation based unique service identifiers. Our design objective of this research is to create virtual representation of entities. Therefore, the dotted service identifiers reflect semantic hierarchy of entities and incorporates source information as directories. Reusable Instance Pool component enables remote programmability of DTs. This component creates a representational instance of DTs upon request from DTIAL component of OpenDT framework. Figure 51 depicts a programming interface to create DT instances that application and mashups can access from code.

There are quite a few models of DTs have been proposed by the researchers. In this chapter, we consider two structural patterns of DTs. First, a DT represents an entire entity such as Renault engine DTs from the motivating scenario from Section II. Second, a DT comprises of multiple sub-product DTs such as Mercedes-Benz cars. Hence, in our formalization, we consider a DT, denoted by DT, consists of one or more sub-ordinate DTs, dt, such that dt $\in$DT. The following equation defines the construction of DT.

$$DT = \{dt_i \; where \; i = 1 \ldots |DT|\} \qquad (4)$$

4.3.3 Service Collection Descriptor model

In this section, we discuss Service Collection Descriptor component and present its information model. We construct an information hierarchy for the proposed Service Descriptor module, denoted by SD_{dt} from. A SD_{dt} comprises of a collection of remote services. According to our proposed design, each DT service a functionality of an entity. Figure 52 outline an information model for DT descriptors.

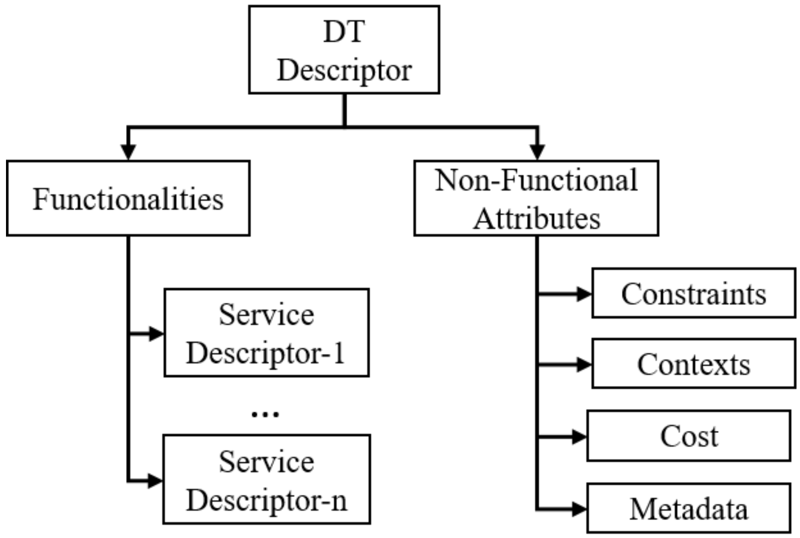


Figure 52: Information model of DT descriptor.

We formalize the outlined DT descriptor model. We consider remote services, denoted by s, consist of three basic information such as input parameters description (in), functional description ($desc$), and output description (out). Hence, $s = \{in, desc, out\}$. Hence, collection of services in a DT can be expressed as $Svc = \{s_i \text{ , } where\ i = 1 \dots |svc|\}$.

DT developers can combine these remote services to create mashups and composite DTs. A mashup service, denoted by ms and transitions between s_i and s_{i+1} is denoted by δ_i, combines multiple svc, can be expressed as $ms = \left\{\prod_{i\geq 1}^{i\leq |svc|} s_i\right\} \cup \{\delta_i, where\ 1 \leq i \leq |ms| - 1\}$. Note that, δ_i is only applicable to ms when output of $s_i$is sequentially fed to any $s_{j\neq i}$. All mashup services in a composite DT can be expressed by $MS = \{ms_i, where\ i = 1 \dots |MS|\}$. Therefore, we can further formalize Svc of a composite DT as $Svc_{composite} = \{Svc, MS\}$.

4.3.4 DT Query model

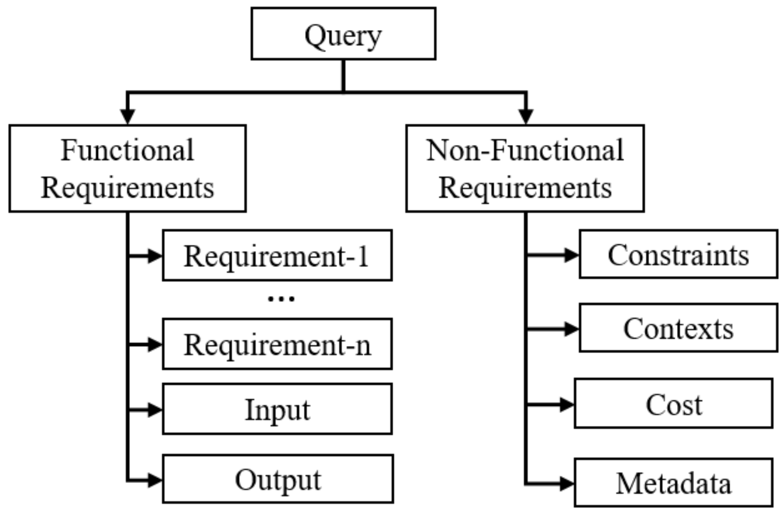


Figure 53: Query data hierarchy in OpenDT.

Query data model is a widely researched topic in service computing. For example, [95] presents a query data model for IoT enabled entities where they incorporate different contexts of service discovery. In this section, we take insights from the existing query models and redesign for OpenDT. Figure 53 depicts our proposed query model for OpenDT.

A query has two types of requirements such as functional requirements and non-functional requirements. Functional requirements consist of a requirement collection. These requirements specify both DT level requirements and service level requirements. Because in OpenDT, the general approach is to discover DTs of entities and then discover suitable services later. Non-functional requirements incorporate required characteristics of entities such as constraints, contexts, operational costs, and other metadata about DTs. DT Discover Manager component of OpenDT crossmatches these functional and non-functional requirements with Service Collection Descriptors of DTs to recommend DTs.

4.4 Implementation

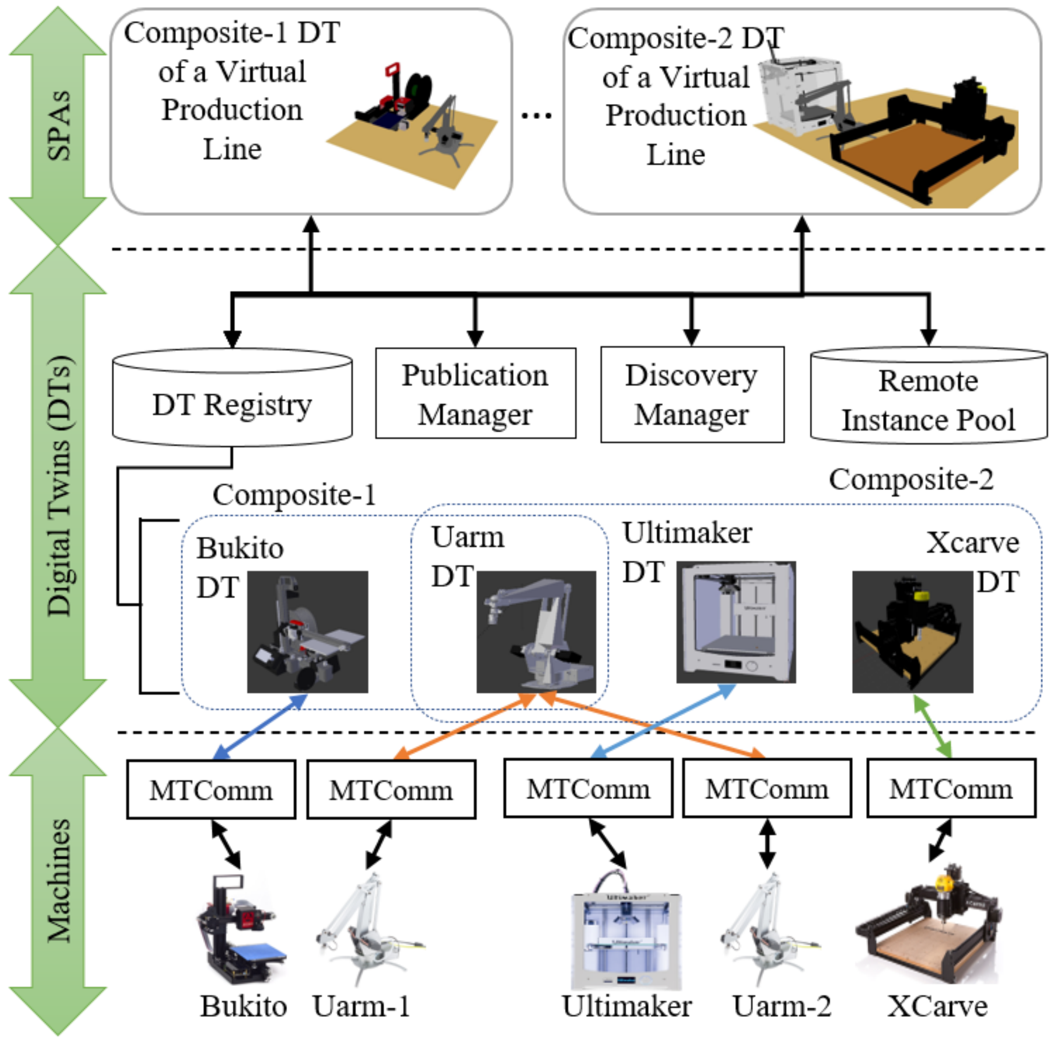


Figure 54: Testbed structure for case validations. All five machines are represented by four DTs. They collaborate in two groups such as Composite-1 (Bukito-Uarm1) and Composite-2 (Bukito-Uarm2-XCarve). Decision support during collaborations are established from the DTs. Established collaborations are event-driven and service-oriented.

We evaluate our framework using a testbed presented in Figure 54. This testbed structure belongs to the domain of cyber-manufacturing. Its infrastructure includes five manufacturing machines such as two 3D printer (Ultimaker2 and Bukito), two robotic arms (Uarm), and one CNC cutting machine (XCarve). We virtualize each machine to their respective cloud-based DTs using MTComm method. MTComm [66] presents an XML-based semantic ontology data organization of cyber-enabled machine tools. In [65], a method was developed to virtualize manufacturing resources as DTs using semantic ontology of MTComm.

All the DTs provide hypermedia web services for near real-time 3D visualization. They also expose web services for remote monitoring and functionalities of manufacturing entities. Each DT offer XML-based Service Collection Descriptor document. Their instances are accessible through the Remote Instance Pool component.

A similar and comparable testbed was implemented in [51] using traditional service publication approaches, namely CPMC testbed. In this section, we contrast the differences between OpenDT testbed and CPMC testbed. For experiments, we have developed multiple Single Page Applications (SPA) named as Ultimaker2, Bukito, XCarve, Composite-1, and Composite-2. These SPAs were developed in both OpenDT and CPMC testbeds.

In CPMC testbed, machine tools are virtualized using web services for monitoring and operations. For Ultimaker2 machine, there were 15 web services published in service registry. Similarly, there were 15, 13, and 6 web services published in service registry for Bukito, XCarve, and Uarm machines respectively. In both testbeds, each of these SPAs discovers services for monitoring and operations on runtime. In CPMC testbed, all the five SPAs were developed where they queried for web services from a traditional service registry.

In OpenDT testbed, SPAs use programmable DT instances from the implemented OpenDT framework. The first three SPAs offer remote machine monitoring by mashing DT services. Composite-1 SPA accesses instances of Bukito DT and UARM DT and configures a new composite manufacturing entity. This entity establishes a collaboration where Bukito prints an object and requests Uarm-1 to move it to storage. Similarly, Composite-2 SPA creates a collaborative manufacturing entity where Ultimaker2 prints an object, requests Uarm-2 to ship it to XCarve for further processing.

4.4.1 Results discussion

4.4.1.1 *Evaluation metrics*

Table 2: List of evaluation metrics

Metric	Motivation
Number of service registries	To evaluate a new publication approach and better discoverability.
Line of Code (LOC)	To evaluate easier programmability with remote programmable DTs.
Number of discovery service calls	To evaluate traffic reduction in runtime service discovery.

We identify three evaluation metrics to perform initial assessment of OpenDT. The following table presents those metrics to contrast service publication and discovery performance in SPAs of CPMC and OpenDT testbed.

4.4.1.2 *Case validations*

Advantages in publication approach: The key research objective of this work is to publish entities with its collection of functionalities, which potentially reduces number of entries in service registry. We draw a comparison between the number of registries created for SPAs in CPMC and OpenDT testbeds. In both of these testbeds, service registry database schema remains mostly identical. However, due to the publication approach, number of registries were reduced significantly in OpenDT testbed. For example, total number of publishable services in machines of Composite-2 DT sums to 34 where Ultimaker2, Bukito and XCarve contributes 15, 13, and 6 respectively. All of these services were needed to be published in CPMC testbed. Contrarily, in

OpenDT framework, we needed to publish only 4 DTs, one for the composite DT and three for the collaborating DTs.

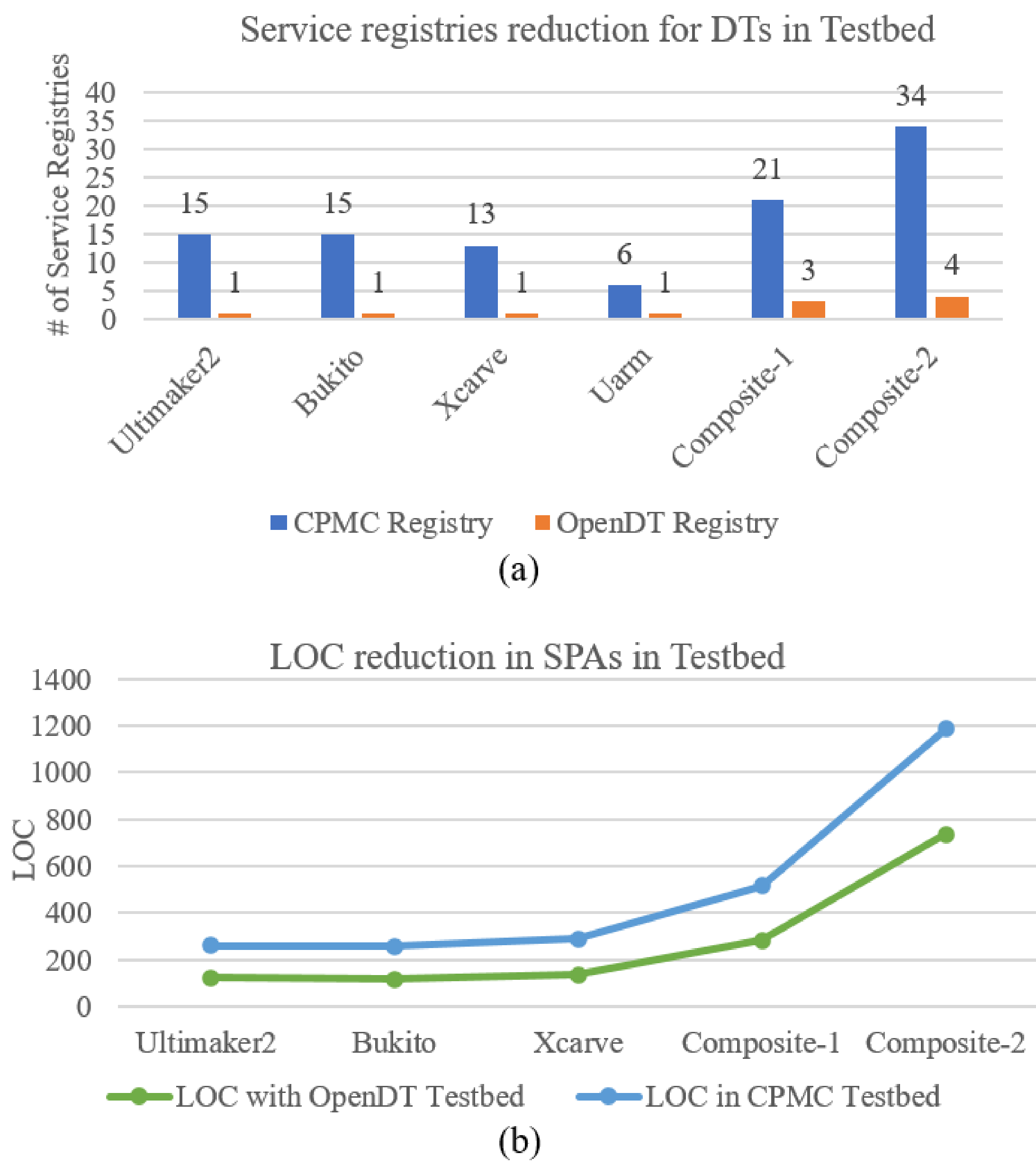


Figure 55: Testbed Results. (a) presents Line of Code (LOC) reduction in SPAs. (b) depicts reduction of service registries using OpenDT.

Advantages in remote programmability: We also draw a comparison of Line of Codes (LOC) among the SPAs developed in CPMC and OpenDT testbed. The SPAs do not need to import jQuery library to invoke DT services. Instead, they invoked services by the DT instances. Our observation shows that building SPAs by using the proposed DTs reduces 38%-53% LOC. This observation proves that developers spend less time writing code when using our proposed OpenDT framework. In addition, they have an easier method to access the DT services directly from their code base.

Advantages in reduced discovery service traffic: In the context of the considered testbeds, we have seen significant reduction of discovery service calls in OpenDT. This significant reduction on service calls is achieved because the DT instances already contains the service collection descriptor. Consequently, the application code avoided making repetitive remote service calls to DT Registry component to discover other functionalities of the entity.

4.4.1.3 *Related works*

Existing service publication and discovery approaches: Service registries or catalogs are useful tools for service publication and management. Traditional service catalog models and methods offers multiple functionalities such as service search, discover, learning, and documentation. A service publication and discovery model for IoT enabled things was presented in [95], where services of things are published in an individual approach and recommended to users based on contexts of the user. A cloud-based IoT mashup as a service model was presented in [96], where a collection of IoT sensors were combined and communicated through a web service. There are commercial management tools of service catalogs for IoT enabled entities such as Bosch IoT Suites, Microsoft Azure IoT Suite, etc. These literatures and technologies provide important insights for service publication and discovery. However, none of these works has proposed any approach that reduces remote service invocation for service discovery.

*Service-oriented architectures of DTs***:** In 2002, Grieves introduced the concept of DT in the domain of manufacturing for production lifecycle management [56]. In 2011, NASA redefined the term DT from the perspective of unmanned vehicular research and development. They defined a DT as an integrated multiphysics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, and others, to mirror the life of its corresponding flying twin [51].

The number of DT representations and application domains are increasing rapidly. However, all variants of DT models have one thing in common, which is data-driven representation. Many existing DT models adopt service-oriented architecture (SoA) to collect real-time sensor data from CPS and IoT enabled systems [97]. Some DT models integrate remote control of physical processes from virtual space using web services [65, 51]. However, these DTs are mostly designed for specific domains and there is a need for a more general architecture for service-oriented entities. Rapid employment of DTs across domains have proven that they are one of the strongest candidates of virtual representations. Our proposed architecture utilizes this potential in OpenDT framework.

4.5 Conclusion

Service computing has been one of the most influential tools to integrate technology with different types of entities across domains. Typically, a modern entity publishes its functionalities as remote services in service registries where each individual service creates at least an entry. Therefore, service registries are getting overpopulated because of increasing number of individually published services, which potentially impedes discovery performance. In this chapter, we take a new publication approach to reduce number of entries in service registries and improve discovery performance. We investigate if one entry in registry can publish all services of an entire entity. Towards this objective, we connect the concept of DT, which can virtually represent entities and offer a collection of services from one entry of registry. To support the proposed idea, we design a DT publication and discovery framework, namely OpenDT. We detail the framework and its essential components including a service-oriented architecture of DTs for heterogeneous entities. We validate OpenDT using a prototype implementation and case validations clearly show early promises of this framework.

This research is at the beginning stage and focuses on fundamental concept development. More research is required in future towards the direction of faster service discovery not only in the contexts of IoT and CPS based entities but also for wider variety of domains. In addition, future research should include a expand OpenDT and evaluated using more large and complex entities. Last but not the least, this framework should be experimented with time-critical entities and applications.

REFERENCES

[1] C. Xu-dong, "Cloud manufacturing:a new service-oriented networked manufacturing model," *Computer Integrated Manufacturing Systems,* 2010.

[2] X. Xu, "From cloud computing to cloud manufacturing," *Robotics and Computer-integrated Manufacturing,* vol. 28, no. 1, pp. 75-86, 2012.

[3] E. A. Lee, "Cyber-Physical Systems-Are Computing Foundations Adequate ?," , 1998.

[4] E. A. Lee, "Cyber Physical Systems: Design Challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC),* 2008.

[5] L. Wang, R. Gao and I. Ragai, "An Integrated Cyber-Physical System for Cloud Manufacturing," in *ASME 2014 International Manufacturing Science and Engineering Conference, MSEC 2014 Collocated with the JSME 2014 International Conference on Materials and Processing and the 42nd North American Manufacturing Research Conference; Detroit; United States*, 2014.

[6] M. M. Helu and T. D. Hedberg, "Enabling Smart Manufacturing Research and Development using a Product Lifecycle Test Bed," *Procedia Manufacturing,* vol. 1, pp. 86-97, 2015.

[7] F. Tao, L. Zhang, V. C. Venkatesh, Y. Luo and Y. Cheng, "Cloud manufacturing: a computing and service- oriented manufacturing model," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* vol. 225, no. 10, pp. 1969-1976, 2011.

[8] D. W. Rosen, D. Schaefer and D. Schrage, "GT MENTOR: A High School Education Program in Systems Engineering and Additive Manufacturing," *23rd Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Symposium (SFF 2012),* 2012.

[9] D. Schaefer, Cloud-Based Design and Manufacturing (CBDM): A Service-Oriented Product Development Paradigm for the 21st Century, 2014.

[10] D. Wu, M. J. Greer, D. W. Rosen and D. Schaefer, "Cloud manufacturing: Strategic vision and state-of-the-art," *Journal of Manufacturing Systems,* vol. 32, no. 4, pp. 564-579, 2013.

[11] B. Esmaeilian, S. Behdad and B. Wang, "The evolution and future of manufacturing: A review," *Scopus,* 2016.

[12] B. Huang, C. Li, C. Yin and X. Zhao, "Cloud manufacturing service platform for small- and medium-sized enterprises," *The International Journal of Advanced Manufacturing Technology,* vol. 65, no. 9, pp. 1261-1272, 2013.

[13] S. Dong and G. Jianling, "A strategy of building Cloud Manufacturing Service Platform based on CloudAnalyst," in *Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on*, 2013.

[14] F. Zhang, Z. Zhou, W. Xu and Y. Zhao, "Cloud Manufacturing Resource Service Platform based on Intelligent Perception Network using Fiber Optic Sensing," *International Journal on Advances in Information Sciences and Service Sciences,* vol. 4, no. 23, pp. 366-372, 2012.

[15] P. Helo, M. Suorsa, Y. Hao and P. Anussornnitisarn, "Toward a cloud-based manufacturing execution system for distributed manufacturing," *Computers in Industry,* vol. 65, no. 4, pp. 646-656, 2014.

[16] U. Rauschecker, M. Meier, R. Muckenhirn, A. L. K. Yip, A. P. Jagadeesan and J. Corney, "Cloud-based manufacturing-as-a-service environment for customized products," in *eChallenges e-2011 Conference*, 2011.

[17] X. V. Wang and X. W. Xu, "An interoperable solution for Cloud manufacturing," *Robotics and Computer-integrated Manufacturing,* vol. 29, no. 4, pp. 232-247, 2013.

[18] L. Wang, M. Törngren and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," *Journal of Manufacturing Systems,* vol. 37, pp. 517-527, 2015.

[19] V. P. Modekurthy, X. F. Liu, K. K. Fletcher and M. C. Leu, "Design and Implementation of a Broker for Cloud Additive Manufacturing Services," *Journal of Manufacturing Science and Engineering-transactions of The Asme,* vol. 137, no. 4, p. 40904, 2015.

[20] J. Mai, L. Zhang, F. Tao and L. Ren, "Customized production based on distributed 3D printing services in cloud manufacturing," *The International Journal of Advanced Manufacturing Technology,* vol. 84, pp. 71-83, 2016.

[21] E. A. Lee and S. A. Seshia, Introduction to embedded systems: A cyber-physical systems approach, Mit Press, 2016.

[22] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia CIRP,* vol. 17, pp. 9-13, 2014.

[23] K. P. Cooper and R. F. Wachter, "Cyber-enabled manufacturing systems for additive manufacturing," *Rapid Prototyping Journal,* vol. 20, no. 5, pp. 355-359, 2014.

[24] K.-F. Seitz and P. Nyhuis, "Cyber-Physical Production Systems Combined with Logistic Models – A Learning Factory Concept for an Improved Production Planning and Control☆," *Procedia CIRP,* vol. 32, pp. 92-97, 2015.

[25] "Authentise INC," [Online]. Available: https://www.authentise.com/. [Accessed 31 May 2020].

[26] N. Liu, X. Li and Q. Wang, "A resource & capability virtualization method for cloud manufacturing systems," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 2011.

[27] R. Khare and A. Rifkin, "XML: a door to automated Web applications," *IEEE Internet Computing,* vol. 1, no. 4, pp. 78-87, 1997.

[28] E. Guerrieri, "Software document reuse with XML," in *Proceedings. Fifth International Conference on Software Reuse (Cat. No.98TB100203)*, 1998.

[29] P. M. Mell and T. Grance, "THE NIST DEFINITION OF CLOUD COMPUTING," *Special Publication (NIST SP) - 800-145,* vol. 23, no. 6, pp. 50-50, 2011.

[30] L. Li, "China's manufacturing locus in 2025: With a comparison of “Made-in-China 2025” and “Industry 4.0”," *Technological Forecasting and Social Change,* vol. 135, pp. 66-74, 2017.

[31] N. Liu, X. Li and Q. Wang, "A resource & capability virtualization method for cloud manufacturing systems," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, 2011.

[32] Y. Z. Zhao, J. B. Zhang, L. Zhuang and D. H. Zhang, "Service-oriented architecture and technologies for automating integration of manufacturing systems and services," in *2005 IEEE Conference on Emerging Technologies and Factory Automation*, 2005.

[33] F. Ameri and D. Dutta, "A Matchmaking Methodology for Supply Chain Deployment in Distributed Manufacturing Environments," *Journal of Computing and Information Science in Engineering,* vol. 8, no. 1, p. 11002, 2008.

[34] A. Vijayaraghavan, W. Sobel, A. Fox, D. Dornfeld and P. Warndorf, "Improving Machine Tool Interoperability Using Standardized Interface Protocols: MT Connect," *Laboratory for Manufacturing and Sustainability,* 2008.

[35] A. Vijayaraghavan and D. Dornfeld, "ADDRESSING PROCESS PLANNING AND VERIFICATION ISSUES WITH MTCONNECT," *Laboratory for Manufacturing and Sustainability,* 2009.

[36] J. L. Michaloski, B. E. Lee, F. M. Proctor, S. Venkatesh and S. Ly, "Quantifying the Performance of MT-Connect in a Distributed Manufacturing Environment," in *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2009.

[37] Y.-L. Lin, C.-C. Lin and H.-S. Chiu, "The development of intelligent service system for machine tool industry," in *2015 1st International Conference on Industrial Networks and Intelligent Systems (INISCom)*, 2015.

[38] VIMANA. [Online]. Available: http://www.systeminsights.com/.

[39] R. Industrial. [Online]. Available: http://rosindustrial.org/.

[40] L. America. [Online]. Available: http://www.lns-america.com.

[41] Mazak. [Online]. Available: https://www.mazakusa.com/..

[42] O3, 12 08 2019. [Online]. Available: http://www.uilabs.org/project/o3-operate-orchestrate-and-originate-14-06-05/.

[43] STEP, 12 08 2019. [Online]. Available: http://www.steptools.com/.

[44] QIF, 12 08 2019. [Online]. Available: http://qifstandards.org.

[45] ROY-G-BIV, "ROY-G-BIV," 12 08 2019. [Online]. Available: http://www.roygbiv.com.

[46] X. F. Liu, S. M. N. A. Sunny, R. Shahriar, M. C. Leu, M. Cheng and L. Hu, "Implementation of MTConnect for Open Source 3D Printers in Cyber Physical Manufacturing Cloud," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2016.

[47] X. F. Liu, M. R. Shahriar, S. M. N. Al Sunny, M. C. Leu, M. Cheng and L. Hu, "Design and implementation of cyber-physical manufacturing cloud using MTConnect," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2016.

[48] S. M. N. A. Sunny, X. F. Liu and R. Shahriar, "MTComm: A Semantic Ontology Based Internet Scale Communication Method of Manufacturing Services in a Cyber-Physical Manufacturing Cloud," in *2017 IEEE International Congress on Internet of Things (ICIOT)*, 2017.

[49] S. M. N. A. Sunny, X. F. Liu and R. Shahriar, "Communication method for manufacturing services in a cyber–physical manufacturing cloud," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 7, pp. 636-652, 2017.

[50] S. M. N. A. Sunny, X. Liu and R. Shahriar, "Remote Monitoring and Online Testing of Machine Tools for Fault Diagnosis and Maintenance Using MTComm in a Cyber-Physical Manufacturing Cloud," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 2018.

[51] X. F. Liu, M. R. Shahriar, S. M. N. Al Sunny, M. C. Leu and L. Hu, "Cyber-physical manufacturing cloud: Architecture, virtualization, communication, and testbed," *Journal of Manufacturing Systems*, vol. 43, pp. 352-364, 2017.

[52] N. Liu and X. Li, "A multilevel modeling framework for semantic representation of cloud manufacturing resources," in *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2013.

[53] Y. Jiming, G. Zhiping and S. Rongbo, "Perception of Manufacturing Resources in Cloud-Manufacturing System," in *2012 International Conference on Computer Science and Service System*, 2012.

[54] H.-K. Lin and J. A. Harding, "A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration," *Computers in Industry,* vol. 58, no. 5, pp. 428-437, 2007.

[55] L. Xiaoqing, A. S. S. M. Nahian and S. M. Rakib, *Semantic Ontology-Based Internet Scale Communication Method of Machine Tools for Providing Remote Operational Services,* 2018.

[56] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary perspectives on complex systems*, Springer, 2017, pp. 85-113.

[57] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *The International Journal of Advanced Manufacturing Technology,* vol. 94, pp. 3563-3576, 2018.

[58] K. Ding, F. T. S. Chan, X. Zhang, G. Zhou and F. Zhang, "Defining a Digital Twin-based Cyber-Physical Production System for autonomous manufacturing in smart shop floors," *International Journal of Production Research,* pp. 1-20, 2019.

[59] MTConnect, "MTConnect," 12 08 2019. [Online]. Available: http://mtconnect.org.

[60] A. Angrish, B. Starly, Y.-S. Lee and P. H. Cohen, "A flexible data schema and system architecture for the virtualization of manufacturing machines (VMM)," *Journal of Manufacturing Systems,* vol. 45, pp. 236-247, 2017.

[61] J. Siderska, "Cloud Manufacturing—The Adoption of Virtual Production Line to Soft Resources Analysis," *International Conference on Engineering, Project, and Product Management,* pp. 225-233, 2017.

[62] M. Yuan, Z. Zhou, X. Cai, C. Sun and W. Gu, "Service composition model and method in cloud manufacturing," *Robotics and Computer-integrated Manufacturing,* vol. 61, p. 101840, 2020.

[63] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang and A. Nee, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems,* 2019.

[64] L. Hu, N.-T. Nguyen, W. Tao, M. C. Leu, X. F. Liu, R. Shahriar and S. M. N. A. Sunny, "Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MT Connect," *Procedia Manufacturing,* vol. 26, pp. 1193-1203, 2018.

[65] M. R. Shahriar, S. M. N. Al Sunny, X. Liu, M. C. Leu, L. Hu and N.-T. Nguyen, "MTComm Based Virtualization and Integration of Physical Machine Operations with Digital-Twins in Cyber-Physical Manufacturing Cloud," in *2018 5th IEEE International Conference on Cyber*

Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2018.

[66] S. M. N. Al Sunny, X. F. Liu and M. R. Shahriar, "MTComm: A Semantic Ontology Based Internet Scale Communication Method of Manufacturing Services in a Cyber-Physical Manufacturing Cloud," in *Internet of Things (ICIOT), 2017 IEEE International Congress on*, 2017.

[67] Y. Lu and X. Xu, "Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services," *Robotics and Computer-integrated Manufacturing,* vol. 57, pp. 92-102, 2019.

[68] K. T. Park, J. Lee, H.-J. Kim and S. D. Noh, "Digital twin-based cyber physical production system architectural framework for personalized production," *The International Journal of Advanced Manufacturing Technology,* vol. 106, no. 5, pp. 1787-1810, 2020.

[69] C. Liu, P. Jiang and W. Jiang, "Web-based digital twin modeling and remote control of cyber-physical production systems," *Robotics and Computer-integrated Manufacturing,* vol. 64, p. 101956, 2020.

[70] P. D. U. Coronado, R. Lynn, W. Louhichi, M. Parto, E. Wescoat and T. Kurfess, "Part data integration in the Shop Floor Digital Twin: Mobile and cloud technologies to enable a manufacturing execution system," *Journal of Manufacturing Systems,* 2018.

[71] H. ElMaraghy and W. ElMaraghy, "Smart adaptable assembly systems," *Procedia CIRP*, vol. 44, pp. 4-13, 2016.

[72] D. Kibira and C. McLean, "Manufacturing modeling methods: virtual reality simulation of a mechanical assembly production line," in *Proceedings of the 34th conference on Winter simulation: exploring new frontiers,* 2002.

[73] C. Zhuang, J. Liu and H. Xiong, "Digital twin-based smart production management and control framework for the complex product assembly shop-floor," *The International Journal of Advanced Manufacturing Technology,* vol. 96, pp. 1149-1163, 2018.

[74] M. Grieves, "Origins of the Digital Twin Concept," *Florida Institute of Technology,* 2016.

[75] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne and L. Wang, "Modeling, simulation, information technology & processing roadmap," *National Aeronautics and Space Administration,* 2012.

[76] E. D. Simmon, K.-s. Kim, E. Subrahmanian, R. Lee, F. J. d. Vaulx, Y. Murakami, K. Zettsu and R. D. Sriram, "A Vision of Cyber-Physical Cloud Computing for Smart Networked Systems," *NIST Interagency/Internal Report (NISTIR) - 7951,* 2013.

[77] K. Ding, F. T. S. Chan, X. Zhang, G. Zhou and F. Zhang, "Defining a Digital Twin-based Cyber-Physical Production System for autonomous manufacturing in smart shop floors," *International Journal of Production Research,* pp. 1-20, 2019.

[78] A. J. H. Redelinghuys, K. Kruger and A. Basson, "A Six-Layer Architecture for Digital Twins with Aggregation," in *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, 2019.

[79] Z. Lin and A. Wang, "3D virtual and real synchronous mapping modeling and operation control technology oriented to automatic test unit," in *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*, 2018.

[80] C. Cimino, E. Negri and L. Fumagalli, "Review of digital twin applications in manufacturing," *Computers in Industry,* vol. 113, p. 103130, 2019.

[81] A. Angrish, B. Starly, Y.-S. Lee and P. H. Cohen, "A flexible data schema and system architecture for the virtualization of manufacturing machines (VMM)," *Journal of Manufacturing Systems,* vol. 45, pp. 236-247, 2017.

[82] Oracle, "Developing Applications with Oracle Internet of Things Cloud Service," Oracle Inc, [Online]. Available: https://docs.oracle.com/en/cloud/paas/iot-cloud/iotgs/oracle-iot-digital-twin-implementation.html. [Accessed 30 10 2019].

[83] ANSYS, "Digital Twin Technology," ANSYS, [Online]. Available: https://www.ansys.com/en-gb/products/systems/digital-twin. [Accessed 30 10 2019].

[84] T. Jäckle, "Eclipse Ditto: How digital twins boost development in the IoT," Bosch, [Online]. Available: https://blog.bosch-si.com/developer/how-digital-twins-boost-development-in-the-iot/. [Accessed 30 10 2019].

[85] X. Yang, G. Shi and Z. Zhang, "Collaboration of large equipment complete service under cloud manufacturing mode," *International Journal of Production Research,* vol. 52, no. 2, pp. 326-336, 2014.

[86] O. F. Valilai and M. Houshmand, "A collaborative and integrated platform to support distributed manufacturing system using a service-oriented approach based on cloud computing paradigm," *Robotics and computer-integrated manufacturing,* vol. 29, no. 1, pp. 110-127, 2013.

[87] N. Gama, M. Mar Rosa and M. M. Silva, "IT services reference catalog," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, 2013.

[88] W. J. Obidallah and B. Raahemi, "A survey on web service discovery approaches," in *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*, 2017.

[89] H. Zorgati, R. B. Djemaa and I. A. B. Amor, "Service discovery techniques in Internet of Things: a survey," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019.

[90] "ProgrammableWeb," ProgrammableWeb, 2005. [Online]. Available: https://www.programmableweb.com/. [Accessed 24 May 2020].

[91] T. Wang, C. Niu and L. Cheng, "A two-phase context-sensitive service composition method with the workflow model in cyber-physical systems," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, 2014.

[92] T. Catarci, D. Firmani, F. Leotta, F. Mandreoli, M. Mecella and F. Sapio, "A Conceptual Architecture and Model for Smart Manufacturing Relying on Service-Based Digital Twins," in *2019 IEEE International Conference on Web Services (ICWS)*, 2019.

[93] Y. Qamsane, C.-Y. Chen, E. C. Balta, B.-C. Kao, S. Mohan, J. Moyne, D. Tilbury and K. Barton, "A Unified Digital Twin Framework for Real-time Monitoring and Evaluation of Smart Manufacturing Systems," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019.

[94] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE access,* vol. 5, pp. 2050-2062, 2017.

[95] N. Ibrahim, "Publication and Discovery of Things in the Internet of Things," *EAI Endorsed Transactions on Internet of Things,* vol. 4, 2018.

[96] J. Im, S. Kim and D. Kim, "IoT mashup as a service: Cloud-based mashup service for the internet of things," in *2013 IEEE international conference on services computing*, 2013.

[97] Y. Lu, C. Liu, I. Kevin, K. Wang, H. Huang and X. Xu, "Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues," *Robotics and Computer-Integrated Manufacturing,* vol. 61, p. 101837, 2020.

[98] M. Online, 12 08 2019. [Online]. Available: https://www.mmsonline.com/articles/bringing-theinternet-to-the-machine-tool.

APPENDIX: ALL PUBLISHED AND SUBMITTED PUBLICATIONS

Chapter 1: X. F. Liu, M. R. Shahriar, S. M. N. A. Sunny, M. C. Leu, M. Cheng, and L. Hu, "Design and Implementation of Cyber-Physical Manufacturing Cloud Using MTConnect," Volume 1B: 36th Computers and Information in Engineering Conference, 2016.

Chapter 1: X. F. Liu, M. R. Shahriar, S. N. A. Sunny, M. C. Leu, and L. Hu, "Cyber-physical manufacturing cloud: Architecture, virtualization, communication, and testbed," Journal of Manufacturing Systems, vol. 43, pp. 352–364, 2017.

Chapter 2: M. R. Shahriar, S. M. N. A. Sunny, X. Liu, M. C. Leu, L. Hu, and N.-T. Nguyen, "MTComm Based Virtualization and Integration of Physical Machine Operations with Digital-Twins in Cyber-Physical Manufacturing Cloud," 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2018.

Chapter 3: M. R. Shahriar, S. M. N. A. Sunny, and X. F. Liu, "Digital Twin based Virtual Production Lines in Cyber-Physical Manufacturing Clouds", 2019. *(Submitted in Special Issue on Digital Twin towards Smart Manufacturing and Industry 4.0 in the Journal of Manufacturing Systems.)*

Chapter 4: M. R. Shahriar, X. F. Liu, M. M. Rahman, and S. M. N. A. Sunny, "OpenDT: A Reference Framework for Service Publication and Discovery using Remote Programmable Digital Twins", 2020. *(Submitted in IEEE SCC 2020)*