International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: https://www.tandfonline.com/loi/tcim20

Communication method for manufacturing services in a cyber–physical manufacturing cloud

S. M. Nahian Al Sunny, Xiaoqing F. Liu & Md Rakib Shahriar

To cite this article: S. M. Nahian Al Sunny, Xiaoqing F. Liu & Md Rakib Shahriar (2018) Communication method for manufacturing services in a cyber–physical manufacturing cloud, International Journal of Computer Integrated Manufacturing, 31:7, 636-652, DOI: 10.1080/0951192X.2017.1407446

To link to this article: https://doi.org/10.1080/0951192X.2017.1407446

INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING
2018, VOL. 31, NO. 7, 636–652
https://doi.org/10.1080/0951192X.2017.1407446

ARTICLE

Communication method for manufacturing services in a cyber–physical manufacturing cloud

S. M. Nahian Al Sunny, Xiaoqing F. Liu and Md Rakib Shahriar

Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR, USA

ABSTRACT

The integration of cyber–physical systems and cloud manufacturing has potential to change manufacturing processes for better manufacturing accessibility, agility, and efficiency. To achieve this, it is necessary to establish a communication method of manufacturing services over the Internet in order to access and manage manufacturing resources from the cloud. Most of the existing industrial automation protocols utilise Ethernet-based Local Area Network (LAN) and are not designed specifically for communication of manufacturing services over the Internet. MTConnect has been gaining popularity as a standard for monitoring status of the machine tools remotely, but it is designed for communication of read-only monitoring data of machine tools. This paper presents an agent–adapter-based communication method of manufacturing services in a cyber–physical manufacturing cloud (CPMC) to enable manufacturing with various physically connected machines from geographically distributed locations over the Internet. The system uses MTConnect for monitoring and HTTP-based communication method for operating manufacturing resources through the cloud. This integrated approach allows machine tools to communicate with each other over the Internet in manufacturing processes. This paper presents design of the agent–adapter architecture of the integrated communication method of manufacturing services and then discusses the system’s capability of conducting collaborative manufacturing using the communication method. A testbed of the CPMC using the communication method is developed and it is described in detail in this paper. Two empirical studies are presented in order to show the performance of the proposed communication method and CPMC in multiple manufacturing scenarios. It demonstrates excellent feasibility and effectiveness of the communication method and cyber–physical manufacturing cloud for manufacturing with machines in geographically distributed locations.

ARTICLE HISTORY
Received 1 November 2016
Accepted 13 November 2017

KEYWORDS
Cyber–physical manufacturing cloud; agent–adapter based communication method; MTConnect; HTTP; collaborative manufacturing

1. Introduction

The rapid growth of the Internet has significant impacts on almost every aspect of human life. The cyber world is becoming more and more integrated with the physical world. Cyber–physical systems (CPSs) are gaining momentum in finding applications in a wide range of domains, including advanced and smart manufacturing. CPSs offer the ability to connect physical devices to the Internet by integrating cybernetics, mechatronics, design, and process science (NSF 2011; Khaitan and McCalley 2015; Hancu et al. 2007). CPS is also similar to the Internet of Things (IoT) sharing the same basic architecture, nevertheless, CPS presents a higher combination and coordination between physical and computational elements (Rad et al. 2015). The potential of CPSs in smart manufacturing is enormous. CPSs are also considered to be the backbone of Industrie 4.0, which is believed to be the future of manufacturing paradigm (Kagermann et al. 2013). However, although numerous manufacturing machines are network compatible nowadays, very few of them are operated in a networked environment, mostly due to lack of standardised communication protocols for connecting machines physically over the Internet. Many manufacturers provide Internet connectivity with their machines. But most of these machines do not utilise their Internet connectivity most of the time and their usage is limited to manufacturer-specific applications. Cloud manufacturing (CMfg), another emerging manufacturing paradigm, can address this issue by providing manufacturing services over cyber space based on integration of advanced manufacturing with cloud computing. CMfg addresses issues of virtualisation and perception of physical manufacturing resources, and providing manufacturing cloud services over the Internet. The CMfg enables sharing of manufacturing machines from multiple facilities and multiple manufacturers for fulfilling collaborative manufacturing demands (Liu, Li, and Wang2011; Tao, Zuo, Da Xu, and Zhang 2014). There exists a number of publications, which broadly discussed various CMfg architectures and enabling technologies at the conceptual level. However, very few researches presented successful implementation of a cloud based cyber–physical manufacturing system. It is highly desirable to integrate cyber–physical system and CMfg to develop cyber physical manufacturing cloud technology in order to transform the manufacturing paradigm. It can not only improve production rate, optimisation and efficiency of manufacturing processes and reduce production cost, but also connect consumers directly with manufacturing processes and resources. Very few researches have been performed on the integration CPS with CMfg.

CONTACT S. M. Nahian Al Sunny ✉ smsunny@uark.edu

To address this issue, an experimental cyber–physical manufacturing cloud (CPMC) was developed (Liu, Shahriar, et al. 2016, Liu et al. 2017). The CPMC is a framework for offering and managing manufacturing services from cloud by directly operating and sharing machines from many locations and monitoring manufacturing processes over the Internet. Unlike most existing cloud based manufacturing platforms, the CPMC allows direct operations of connected manufacturing machine tools from the cloud. One of the key barriers of constructing such a system is to develop a communication method between the cloud applications and machine tools over the Internet as no existing industrial communication standard or protocol can support this type of operations. Recently, MTConnect (MTConnect Institute 2016) is becoming a widely accepted communication standard for collecting monitoring data from machine tools. But MTConnect only offers monitoring capabilities, it cannot be used to operate machines remotely. To overcome this issue, this paper presents a unique communication method that has been used to develop the CPMC. The method supports both monitoring and operation of machine tools remotely over the Internet. It uses the MTConnect standard for collecting machine data through standard XMLs. Remote machine operations are performed through HTTP based communication. This communication method not only facilitates communication between the cloud applications and machines, but also supports interoperability between different types of machines located in multiple locations from multiple manufacturers. As the machines can communicate with each other over the Internet, the CPMC is also capable of providing collaborative manufacturing services both inside and across organisational boundaries. This communication method was implemented in a CPMC testbed (Liu, Shahriar, et al. 2016). This paper presents the implementation details of the communication method in the testbed of the CPMC and several case studies to evaluate its performance in multiple scenarios.

This paper has four primary contributions: (1) design and development of an agent–adapter based communication mechanism between components of the CPMC; (2) development of methods to enable operation of manufacturing tools over the Internet; (3) development of communication method between multiple machines of different types over the Internet to facilitate collaborative manufacturing in the CPMC; and (4) implementation of a fully operational testbed of the CPMC based on the developed communication method.

The rest of the paper is organised as follows. The next section discusses the related literature survey. Section 3 contains a brief introduction to the Cyber Physical Manufacturing Cloud (CPMC) framework. Section 4 describes the functionalities and specifications of the communication components. Section 5 elaborates the communication mechanisms between the cloud services and the manufacturing resources. The process of communication between machine tools over the Internet is presented in Section 6. Section 7 discusses the method of collaborative manufacturing in the CPMC. The implementation of a functional testbed of the CPMC is described in Section 8. Section 9 consists of some empirical case studies of the CMPC working in different manufacturing scenarios. Discussions and future works are presented in Section 10. Conclusions are given in Section 11.

2. Literature review

2.1. *CMfg* and CPSs

CMfg is an emerging manufacturing technology which makes manufacturing resources and services available over the internet by integrating cloud computing, big data analysis, IoT, and manufacturing (Li et al. 2010; Tao et al. 2011; Xu 2012). Several architectures of CMfg have been proposed by applying cloud computing to manufacturing (Tao et al. 2011; Luo et al. 2011; Xu 2012; Holtewert et al. 2013). Wu et al. (2013) reviewed a few CMfg architectures in their paper. The architectures that were discussed focus on virtualisation of manufacturing resources and services and offer them as online services to consumers. Tao et al. (2011) proposed a CMfg architecture consisting of 10 layers. Wu et al. (2012), (2015)) proposed a method of cloud-based design and manufacturing which enables sharing of manufacturing resources as cloud services, similar to infrastructure as a service (IaaS) or software as a service (SaaS). Wang (2013) proposed a tiered architecture to service oriented manufacturing and connected it to a shop-floor environment to enable real time availability and monitoring. He also discussed machine availability monitoring and machining process planning towards CMfg and mentioned that closed-loop information flow makes process planning and monitoring feasible services for the CMfg. Rauschecker et al. (2011) proposed a uniform representation of manufacturing resource and services across multiple service providers in CMfg. Li, Liu, and Xu (2012) discussed heterogeneous systems and integration access method in CMfg and proposed a solution for the adaptation of manufacturing equipment based on fiber grating sensing technology. Xiang and Hu (2012) and Tao, Cheng, et al. (2014) discussed an intelligent approach for perceiving and accessing manufacturing resources in CMfg using IoT-based technology. Mai et al. (2016) proposed a framework for 3D printing service platform for CMfg.

CPSs are systems of collaborating computational entities which are in intensive connection with the surrounding physical world and its on-going processes, providing and using, at the same time, data-accessing and data-processing services available on the Internet (Hellinger and Seeger 2011; Monostori 2014; Monostori et al. 2016). In most CPSs, various embedded devices are networked to sense, monitor and actuate physical elements in the real world. CPS is being applied in a wide range of domains including advanced manufacturing. Although there exist numerous physical manufacturing machines which are network-ready, very few of them are operated in a networked environment. Cyber–physical production systems (CPPS) consist of autonomous and cooperative elements and subsystems that are connected based on the context within and across all levels of production, from processes through machines up to production and logistics networks. CPPS, relying on the latest and foreseeable further developments of computer science (CS), information and communication technologies (ICT), and manufacturing science and technology (MST) may lead to the fourth industrial revolution, frequently noted as Industrie 4.0 (Kagermann et al. 2013). Within global supply networks, machinery, warehousing systems and production facilities will incorporate in the shape of CPPS. These systems will autonomously exchange information, triggering actions and controlling each other independently within a so called Smart Factory (Zuehlke 2010).

2.2. *Industrial communication methods and standards*

Interoperability and automation of machine tools have always been a big concern for manufacturers. In the early twentieth centuries, the mechanical technology and analog devices were the primary components of the process control systems and manufacturing systems. In 1970s, Programmable Logic Controller (PLC) with limited control functions was introduced which replaced the conventional relay based control systems (Erickson 1996). With the development of digital computers, the scenario changed radically. Numerically Controlled (NC) machines came into play and labyrinths of mechanical linkages were substituted by point-to-point wiring. But this created a new difficulty. Optimised communication networking among different machines in the factory floor became a necessity. In 1985, Fieldbus systems emerged to reduce the complexity of conventional point-to-point wiring systems by connecting digital and analog devices to central controllers (Thomesse 2005; Zurawski 2014). Because of being an open protocol, many Fieldbus systems were developed in parallel and today there exist a number of variations. Over the past two decades, Fieldbus systems have gone through a lot of modifications and become standardised, although not unified. PROFIBUS is considered to be the most successful fieldbus technology and is widely used in industrial automation systems including factory and process automation. As of 9 September 2016, PROFIBUS & PROFINET International (PI) group indicates on its website that PROFINET offers digital communication for data processing and transmission with speeds up to 12 Mbps and supports up to 126 addresses. Control Area Network (CAN) bus is a high-integrity serial bus system which was fundamentally designed to be an automotive vehicle bus (Tindell, Hansson, and Wellings 1994). CANopen and DeviceNet are higher level protocols standardised on top of CAN bus to allow interoperability with devices on the same industrial network (McFarlane 1997). Modbus is a simple, robust and openly published, royalty free serial bus protocol that connects up to 247 nodes (Modbus Organization 2006). Modbus is easy to implement and operate on RS-232 or RS-485 physical links with speeds up to 115K baud. CC-Link was originally developed by Mitsubishi and is a popular open-architecture, industrial network protocol in Japan and Asia (Wikipedia 2014).

Although the Local Area Networks (LAN) based on Ethernet, as part of the TCP/IP and User Datagram Protocol (UDP) stack, rapidly gained much popularity for home and office use, it initially did not gain much acceptance in the industrial automation domain. However, advances in Ethernet technology have made the LANs more suitable to industrial use. The increased data rates of newer Ethernet standards (for example 802.3u Fast Ethernet) made it easier to create real-time Ethernet protocols. The implementation of full-duplex Ethernet lines allows data transmission and reception to occur simultaneously, simplifying bus arbitration difficulties. Introduction of switched networks has allowed Ethernet to be more acceptable for industrial use as opposed to the older hub based networks. The introduction of Ethernet into the field of industrial networking also presented some new challenges. The existing Ethernet standards needed to be extended or modified to meet the rigorous requirements of industrial networks. This was achieved at various levels of the IP stack, and using various approaches. Backwards compatibility with existing fieldbus protocols was also an issue. Many of the newer Ethernet-based fieldbus protocols are extensions of existing protocols and various compatibility philosophies have been implemented. These are classified into four categories by (Sauter 2010). The first is full compatibility at higher layer protocols, such as exists with High-Speed Ethernet (HSE): Emerson, Foundation Fieldbus (Fieldbus Foundation 2001); Modbus/TCP: Schneider (Swales 1999), and Ethernet/IP: Rockwell (Brooks 2001). This approach is especially prevalent in building automation fieldbuses. HSE is implemented on top of the TCP/IP stack, with additional use of standard IP interfaces such as dynamic host configuration protocol and simple network management protocol (Vincent 2001). Modbus is an application layer messaging protocol for Client/Server based communication between devices connected via different types of buses or networks. The Application Layer Protocol Data Unit (APDU) of MODBUS (Function Code and Data) has been encapsulated into an Ethernet frame. Ethernet/IP uses a Common Industrial Protocol (CIP) (ControlNet International and open DeviceNet Vendor association 2001). CIP represents a common application layer for all physical networks of Ethernet/IP, ControlNet and DeviceNet. Ethernet/IP uses the standard Ethernet and switches, thus it can have an unlimited number of nodes in a system. Another approach is compatibility of data objects and models, such as is the case with PROFINET: Siemens, PNO. This method uses proxy hardware to allow communication between the fieldbus media. PROFINET has three different classes – PROFINET Class A, PROFINET Class B, also referred as PROFINET Real-Time (PROFINET RT), and PROFINET Class C or PROFINET IRT (Isochronous and Real-Time). A lesser amount of compatibility is offered through the use of application layer profiles from existing protocols, as is implemented in Ethernet Powerlink: Bernecker and Rainer (IEC 2004f; Zurawski 2014), EtherCAT: Beckhoff, and SERCOS III (IEC 2004e; 2004g; Neumann 2007). Ethernet POWERLINK is implemented on top of IEEE 802.3 and, therefore, permits a free selection of network topology, cross connect, and hot plug. It utilises a polling and time slicing mechanism for real-time data exchange. Such a system is appropriate for all kinds of automation systems ranging from PLC-to-PLC communication down to motion and I/O control. EtherCAT is a protocol that is optimised for data processing using standard IEEE 802.3 Ethernet Frames. Each slave node processes its datagram and inserts the new data into the frame while each frame is passing through. EtherCAT is the MAC layer protocol and is transparent to any higher level Ethernet protocols such as TCP/IP, UDP, Web server, etc. SERCOS III is the third generation of Serial Real-time Communication System (Sercos). It accumulates on-the-fly packet processing for delivering real-time Ethernet and standard TCP/IP communication to deliver low latency industrial Ethernet. Finally, completely new protocols have been developed for Ethernet that have no relationship with any existing protocols and have forgone any compatibility. Examples of such protocols are Ethernet for Plant Automation and Time-Critical Control Network (TCNet) (IEC 2004h). Modbus/TCP (Swales 1999), PROFINET IO (IEC 2004a), and Ethernet/IP with Time synchronisation (IEC 2004c) are also noteworthy Ethernet based Fieldbus protocols.

OLE for Process Control (OPC) is a significant of many manufacturing networks at higher levels by offering a standardised interface for communication of industrial data. Maintained by the OPC Foundation, The OPC specification has combined object linking and embedding (OLE), component object model (COM), and distributed component object model (DCOM) technologies developed by Microsoft (Leitner and Mahnke 2006). The OPC specification outlined a standard set of objects, interfaces, and methods for use in process control and manufacturing automation applications to facilitate interoperability. OPC data access (OPC DA) is the most commonly used OPC specification, which is used to read and write real-time data. It allows real-time communication of process values over Ethernet with a client-server model. Several other variants of OPC have also been developed, including OPC historical data access which permits for acquiring stored values, OPC data exchange for two-way communication using a server-server model and OPC XML Data Access, which uses XML for communication. Later in 2006, the OPC Unified Architecture (OPC UA) has been specified and was being tested and implemented through its Early Adopters program. OPC UA (IEC 62541) combines the functionality of the existing OPC interfaces with new technologies such as XML and Web Services to deliver higher level manufacturing execution system (MES) and enterprise resource planning (ERP) support. OPC and OPC UA provided the opportunity of accessing machine tool not only from factory floor but also from outside the factory. In recent years, MTConnect has acquired much acknowledgements after the release of its version 1.0 in 2008 (Vijayaraghavan et al. 2008). MTConnect is designed to enhance interoperability of manufacturing machines by providing a uniform XML-based data reporting structure. It is fundamentally a read-only framework, i.e. its principal focus is data monitoring and analysis. MTConnect enables manufacturing machines to be monitored over the Internet. The primary objective of MTConnect is to create a universal machine language that is understandable to all machines and also to the users. MTConnect provides a RESTful interface – there is no need of establishing any session or logon/logoff sequence to acquire data. As MTConnect is not designed for any specific type of machines, several types of manufacturing resources such as CNC machine, industrial robot, milling machine, 3D printer (Liu, Sunny, et al. 2016) currently are made compatible with MTConnect standard. In 2010, The OPC Foundation and the MTConnect Institute declared a cooperation to ensure interoperability and consistency between the two standards (ThomasNet 2010). AutomationML (Automation Markup Language) is another promising upcoming open standard series (IEC 62714) for the description of production plants and plant components (Drath et al. 2008). AutomationML describes the contents – what is exchanged between the parties and systems involved. It helps to model plants and plant components with their skills, topology, interfaces, and relations to others, geometry, kinematics, and even logic and behaviour. A joint working group of the AutomationML e.V. and the OPC Foundation deals with the creation of a companion specification ‘AutomationML in OPC UA’ (Henssen and Schleipen 2014).

2.3. Recent advancements in manufacturing

With rapid advancement of IoT devices, connecting machines to the Internet and accessing them remotely are becoming easier. Specially, in case of 3D printers, there exist several applications like OctoPi, Makerbot Mobile, which gives exclusive remote access to the machines. Nowadays several online marketplaces like Shapeways, Sculpteo, i.materialise, 3D Hubs, Ponoko, etc., have been offering cloud based Additive Manufacturing (AM) services (Rayna, Striukova, and Darlington 2015). Shapeways offers 3D printing services of producing a vast range of objects in various types of material. Users can also upload their own designs and have those printed and delivered by Shapeways. Instead of building the manufacturing part itself, Ponoko utilises of a network of manufacturers from all over the world. Clients can submit their designs and Ponoko searches for a suitable registered manufacturer closest to the client who can produce that particular object. i.materialise provides a 3D printing lab alongside the 3D printing service.

Several projects and applications have emerged where MTConnect based data are collected from manufacturing machines and used for data analytics. Xu (2012) proposed the use of MTConnect for CMfg. Michaloski et al. (2009) has shown MTConnect as a viable communication standard for real-time data operations. An alternative of MTConnect standard, TMTC (Taiwanese Machine Tool Connect), has been proposed by Lin, Lin, and Chiu (2015). 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. System Insights with ROS-Industrial developed a peer-to-peer inter machine communication architecture using MTConnect which is used for the communication between robotic arms and Computer Numerical Machines (CNC) (Liu et al. 2017). Another peer-to-peer communication was established by LNS America in collaboration with Mazak Corporation using MTConnect. 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 (UI Labs 2017). The O3 project is still seeking for a solution by combining the data of STEP, Quality Information Framework (QIF), and MTConnect standards into a unified stream that can be used to evaluate the overall quality of a product (Liu et al. 2017). Some other communication mechanisms were also used in several other projects besides MTConnect, such as – Transmission Control Protocol (TCP). Wang, Gao, and Ragai (2014) 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 (ROY-G-BIV 2017).

Another emerging paradigm in manufacturing world is cloud based data analytics. Recently, Siemens has showcased a cloud based open operating system called 'MindSphere', which is a comprehensive data hosting platform designed to collect large amounts of raw data, analyse that data through a range of apps, transform that data into knowledge, and leverage that newly created knowledge to continuously optimise a manufacturer's production assets and business key performance indicators (Siemens 2017). General Electric (GE) has introduced 'Predix' as a cloud based platform-as-a-service (PaaS) to enable industrial-scale analytics for asset performance management and operation optimisation (Wikipedia 2017). Predix provides its customers with open standards to connect their machines and platform to develop and deploy their own apps for data acquisition, analysis and management. Some other companies like 42Q, Plex systems, etc., are providing multitenant cloud based Manufacturing Execution System (MES) services (Neil 2016).

In summary, interoperability of manufacturing resources and automation for manufacturing processes has always been of significance in the field of manufacturing. Researchers have developed many standards, protocols and methods to fulfil this purpose over the years. Many protocols have focused on communication across factory floor over LANs. The more recent communication method MTConnect focused on connection of manufacturing resources over the Internet. But MTConnect provides only a capability of acquiring data from manufacturing resources instead of operating them over the Internet. It is not sufficient to develop a Cyber Physical Manufacturing Cloud (CPMC) by integrating CPS and CMfg (Liu, Shahriar, et al. 2016, Liu et al. 2017). To make the CPMC fully operational, a communication method of operating machines across the Internet as well as monitoring them is needed. This paper presents a communication method used in the CPMC to operate machine tools remotely and increase the interoperability of different types of machines over the Internet.

3. Cyber–physical manufacturing cloud (CPMC)

A conceptual framework of CPMC is presented in Figure 1(a) (Liu, Shahriar, et al. 2016, 2017). 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. CMfg 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. A controller component works as a gateway between the local server and the machine network. The controller receives instructions from the cloud and sends machine data to the cloud. Manufacturing machines inside the factory floor can have any type of network and infrastructure, as the cloud does not communicate with the manufacturing resources directly. 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.

Figure 1(b) 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 grey thick doubly pointed arrows. This architecture consists of four layers. At the bottom, the resource layer contains manufacturing machines maintained by the manufacturers. Manufacturers can offer manufacturing services using these manufacturing resources based on publication-subscription model in service-oriented architecture for the CPMC. Resource virtualisation layer is used to virtualising manufacturing resources and represent them in CPMC. CPMC virtualises manufacturing resources by publishing RESTful web services to be accessible over the Internet. The core cloud layer hosts a set of basic cloud services such as user subscription manager, security manager, and manufacturing virtualisation services repository 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 REST protocol. 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. Each application can provide one or more manufacturing services.

As this paper focuses on the communication method of the CPMC, the details about the architecture and development of the cloud layers and the methods of virtualisation of machines are not discussed further. Liu et al. (2017) has elaborately discussed the development and implementation of a scalable service-oriented architecture of the CPMC. The presented communication method is between the resource layer and the resource virtualisation layer.

4. Agent–adapter-based communication method

Developing a fully operational CPMC system with different sorts of manufacturing machine tools has one major hindrance. Different machine tools use different machine specific languages. Both monitoring data and operational commands vary from machine tool to machine tool. Moreover, some machines use manufacturer specific proprietary languages and communication protocols. This makes the communication both among the machine tools and between the machine tools and cloud services very complex. In order to address this issue, an agent–adapter-based communication method is designed which supports both status monitoring and operations of the machines over the Internet. The communication method communicates with the manufacturing machines in

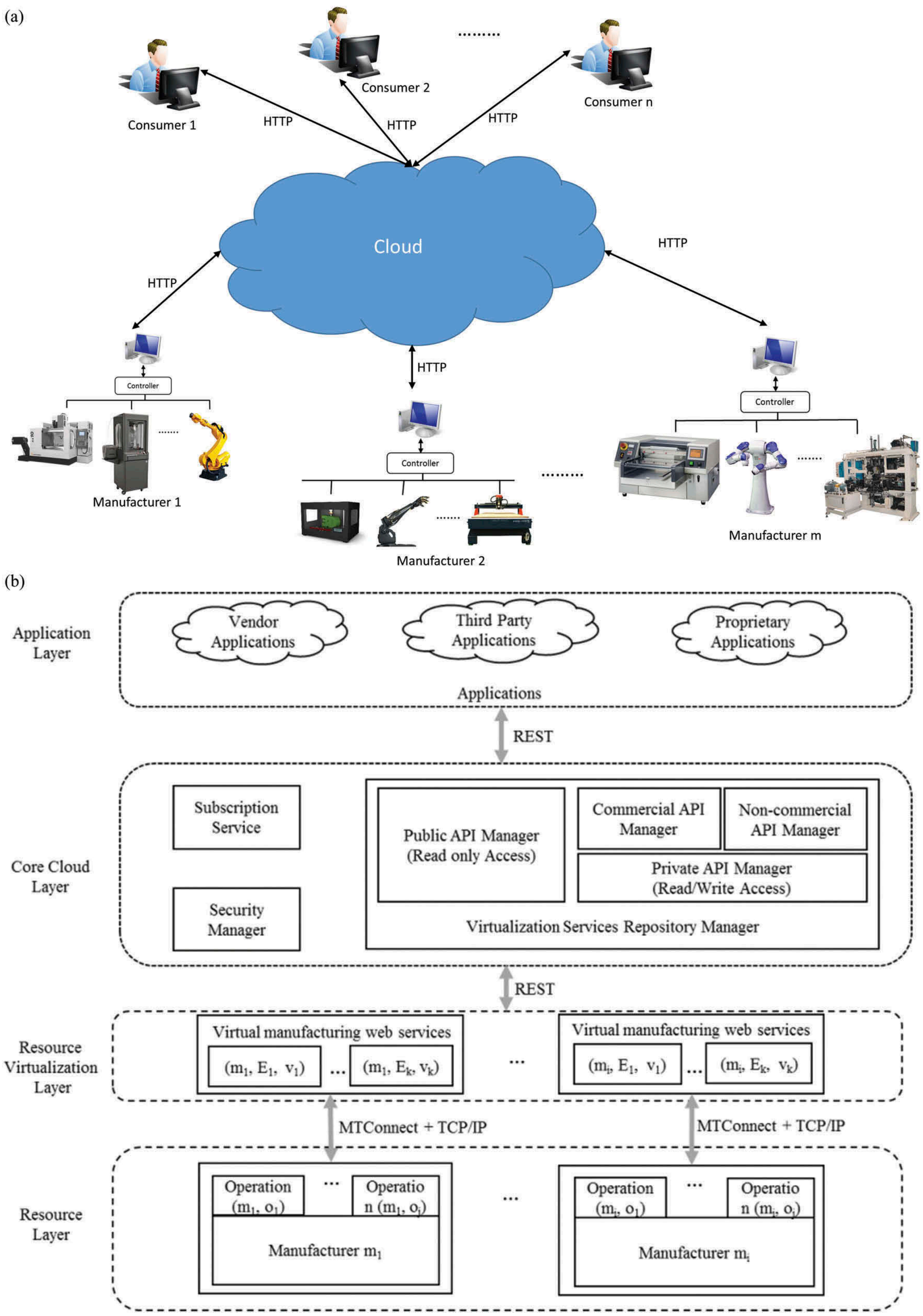


Figure 1. (a) Conceptual framework of CPMC. (b) Four-layer CPMC architecture.

their own specific languages and communicates with other machine tools and the cloud applications in XML and JSON format. The adapter is the operator of the machine and the agent works as the bridge between the adapter and the Internet. Nowadays many modern manufacturing resources operate based on a specific type of file containing the instruction sets, tool paths or models for that particular procedure. For example, to do a drilling job in a typical CNC machine, a G-code file containing the path information is usually sent to the machine. In case of a 3D printer, it takes either a 3D model

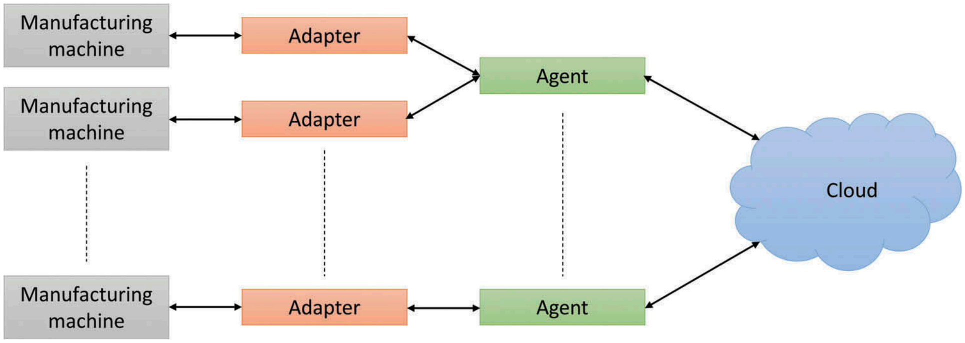


Figure 2. Communication components between the cloud and the machines.

file like STL, VRML file or a G-code file. Most robotic arms work following specific path information written on a given file. Keeping this in mind, the method for conducting machine operations is designed primarily for model file based manufacturing processes and some basic functions like resetting the machine, stop the current process etc. Together the agent and the adapter construct the controller of the machine mentioned in Section 3. The organisation of the agent and the adapter in the system is shown in Figure 2.

```
'availability'          :        'busy'
'x axis position'       :        '92.7'
'y axis position'       :        '-12.2'
'z axis position'       :        '152.3'
'nozzle temperature'    :        '212.5'
'heatbed temperature':           '59.2'
'process progress'      :        '30.0%'
'estimated time'        :        '55 minutes'
```

Figure 3. Sample key-value pair based text dictionary.

4.1. *The adapter*

The adapter is directly connected to the manufacturing resources. It works as a gateway for the machine. Every data flowing inward and outward of the machine goes through the adapter. The role of the adapter is very important as it not only controls and collects data from the machine, but also prevents from direct access to the machine from outside as it only accepts specific sets of commands. Each machine requires its own adapter as different machines' operation principles and mechanisms are different. Adapters are custom-written because the meaning, units and values of data usually differ from machine to machine and device to device. The adapter can be implemented as a software application, or as a combination of software and hardware if the associated machine requires special hardware for data collection or operation. The interface between adapter and machine can be over TCP/IP, RS-232, RS-485, serial I/O etc. The adapter requires access to the machine's core system in such a way through which it can collect data from the machine and also operate the machine.

For reporting the status of the machine, the adapter collects raw monitoring data directly from the machine in machine specific language. The numbers and types of data vary depending on what data the machine can provide and what data the manufacturer wants. Some common data types are machine's availability, the position of the axes, temperatures, progress rate, estimated time of the ongoing process, etc. The data collection process is also dependent of machine type. For example, most 3D printers and CNC machines understand G-code, therefore their adapters send G-code based query and collect the responses. After acquiring the data from the machine, the adapter converts the collected data into a simple key-value pair based text dictionary (Figure 3). The importance of this conversion is twofold. Firstly, it guarantees that different types of data from different machines become consistent and are presented to the agent in one common structure. This facilitates the possibility of using one generic agent for all machines. Secondly, the dictionary makes it easier for the agent to convert the available data into XML format. Once the dictionary is created, it is forwarded to the agent for additional processing.

The process of operating machine tools using the adapter is in a reverse order. The adapter receives operation requests from the agent. To perform the operation, the adapter either sends corresponding commands to the machine or executes a program. For file based operation, the adapter gets the model file from the agent. The data acquisition and operation of machines are performed in parallel.

4.2. *The agent*

The agent provides the interface between the machines and the Internet. As shown in Figure 2, one agent can support multiple adapters. As the data from different machines are translated by their adapters into a common format, one agent can be used for multiple types of machines. The agent's job is divided into two parts – it works as a translator and an Internet gateway for the machines. The adapter sends the key-value pair based text dictionary of status data to the agent. The agent converts this data dictionary into XML format. This function follows the MTConnect standard. MTConnect provides a uniform XML-based data reporting structure for status reporting of machines. The agent uses the MTConnect schemas to convert the data dictionary into MTConnect standard XML format. The agent supports three types of requests – *'probe'*, *'current'*, and *'sample'*. A *'probe'* request solicits information about the physical attributes of the manufacturing equipment and data items that can be retrieved. This helps

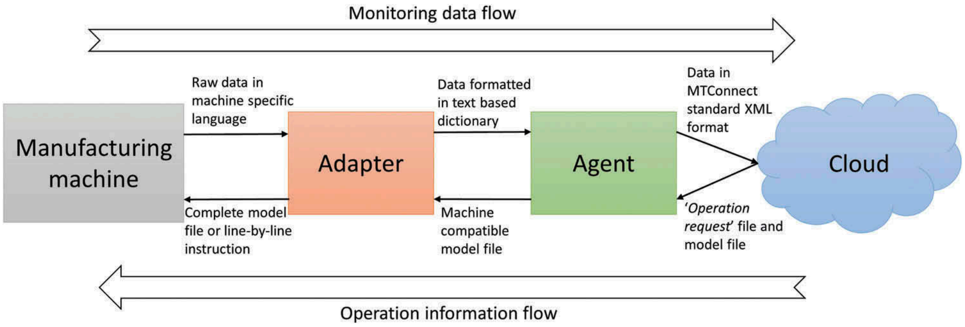


Figure 4. Complete functionality of the agent and adapter in the CPMC.

the resource virtualisation layer of the CPMC to virtualise the manufacturing services associated with the machine and also the application layer to set parameters for data exchange. The other requests are *'current'* which reports back the current state of the machine, i.e. the current value of each attributes, and *'sample'* which provides sample values for a certain period of time. The agent must have sufficient buffer storage to store data until applications request for it. The size of the buffer differs to satisfy the requirements of the implementation. The agent is a software program. Adapters use TCP/IP protocol to communicate with agents.

To support network access and provide monitoring data over the Internet, the agent hosts a RESTful HTTP server to receive requests over the Internet, process those requests and send appropriate responses. The agent uses RESTful protocol – meaning it is stateless on the server side. Functionalities of agents are provided as RESTful web services. The incoming requests for monitoring data are handled through HTTP GET method. When the cloud applications request for monitoring data, the agent's server processes the request, identifies the request type, converts the corresponding data into XML format, and then responds with the formatted data.

For performing machine operations, the operation requests are made to the agent over the Internet. Before starting an operation, the incoming requests are verified to make sure that they are meant for the associated machine. To do so, some parameters must be set and checked by the agent in order to ensure machine's safety. Otherwise, hazardous situation may occur. Therefore, the first step to initiate an operation is to send some specific information about the operation as an '*operation request*'. It includes the ip address of the agent, type of operation, parameters for the operation, and link to the model file required (for file-based operation). With the i.p. address and operation type, the agent verifies whether the operation is directed to this machine and the operation is supported respectively. Additional parameters can be added if necessary. The '*operation request*' is sent as a JSON file. The agent receives the file through the HTTP POST method. Whenever a POST request is made to the agent's server, it analyses and validates the request against the machine's information. If everything is appropriate, the agent checks whether the machine is currently idle or busy in another operation. If busy, it rejects the operation request and sends back acknowledgement. If idle, the agent goes ahead with the process and uses HTTP GET method to download the model file from the given link. Once the file is downloaded and stored, the agent sends command to the adapter to start a new operation and forwards the model file. For other non-file based operations like 'STOP' or 'RESET', the agent gets the operation type from the '*operation request*' file and sends appropriate commands to the adapter. The rest of the process is handled by the adapter. In the meantime, the adapter keeps getting latest status data from the machine and the agent continues to store data in XML format. Figure 4 illustrates the complete work flow of the agent and the adapter.

In brief, the agent and the adapter enable the machine's ability to communicate with the cloud applications over the Internet. The responsibilities of the agent and the adapter are mutually exclusive – the adapter handles communication with machine and the agent talks to the Internet. Both the agent and the adapter provide a lot of flexibility for designers as the method only specify the way of communication, not the way the agent and the adapter should be implemented. This is to make the method more robust and scalable so that it can be made compatible with various kinds of machines.

5. Communication between manufacturing machines and the cloud applications over the internet

In the CPMC, the communication between the resource layer and the resource virtualisation layer is done using the agent–adapter based communication method described in the previous section. The virtualisation services collect monitoring data of a registered machine from its agent through the local server. When a machine is registered with the CPMC, the web services of that machine are added in the cloud's service repository. It contains necessary information for communication, i.e. URL of each service. Manufacturers provide applications in the CPMC for their available manufacturing services. For instance, let a manufacturer have a 3D printer that can produce 3 types of PLA objects. So there is an application in the CPMC for that manufacturer which provides interface to monitor the machine using its monitoring web services and also to place order for producing any of the available objects via its operation services.

For data acquisition the cloud applications make *'current'* requests to the agent of a machine. When a user requests for the current status of a particular machine through its application in the CPMC, the application calls the corresponding web service. The agent of the machine responds with the available data in XML format. Then the core cloud layer processes the data and forwards the data to the application layer which presents the data to the users in a tabular format. The virtualisation services use the *'probe'* request to access the characteristics data of a machine. *'Sample'* request is used for collecting status data for a certain time period.

For operating the machines in the CPMC, a cloud application sends operation request with parameters to the agent of the associated machine to initiate an operation. The clients can browse through applications of the CPMC providing available manufacturing services. When an order is placed, the application stores the model file (either uploaded by the client or generated by other applications in the cloud) in the cloud's repository. As the cloud application has access to the associated web services from the service repository, it calls the monitoring services of the machine to check its availability. Once available, an '*operation request*' file in JSON format is generated containing the address of the machine (IP address and port number), type of operation, type of model file, the link to the model file, and parameters (if any). After that, the file is sent through a POST request to the server of the agent of the selected machine. Once the file is received, the agent of the machine verifies whether this file is meant for it by checking the address stated in the file. It also makes sure that the file type is correct for this machine or not. Then it again checks the machine's availability. Here, the multi-step verification is crucial, as it makes sure that the correct type of file is sent to machine only when the machine is not doing any other job. It prevents the machine for damages and accidents. In any case of verification failure, the agent reports back with type of failure and the cloud services take necessary steps to handle the error. If everything is appropriate, then the agent requests for the model file from the cloud repository using the link given in the '*operation request*' file and downloads the model file into its storage. The procedure is illustrated by a flow chart in Figure 5.

The rest of the operation procedure is handled by the adapter, as described above. For non-file based operations, the '*operation request*' file only contains the address of the machine and the type of operation. When the agent sees one of these type in the '*operation request*' file, it forwards the request to the adapter after verification. Then the adapter sends appropriate commands to the machine to perform the specified operation. There are four basic types of non-file based operations – 'PAUSE', 'RESUME', 'STOP', and 'RESET'. The functionalities of these commands are as their name suggest. 'PAUSE' and 'STOP' can only be performed when the machine is busy in doing a job. 'RESUME' is only applicable when the machine is in paused state. 'RESET' can be used in either idle or busy state, to command the machine to go back to its initial idle state. Additional operations can also be added if necessary.

6. Communication between machine tools over the internet

An important feature of the presented communication method is that it enables the capability of communication between various machine tools in the CPMC over the Internet. Effective machine communication across the factory floor can speed up the production process and can increase machine utilisation by 10–50% (Modern Application News 2016). The CPMC provides a robust manufacturing platform using communication between different types of machine across the Internet. As monitoring data of a machine is available in XML format over the Internet, manufacturing machines can monitor the status of other machines. In addition to that, machines can also initiate operations of other machines using the HTTP based communication method of the CPMC discussed in Section 4. All communications are actually carried out by the agents of the associated machines.

If a machine (M_1) requires service of another machine (M_2), its agent inspects the current status of M_2 from its data in XML

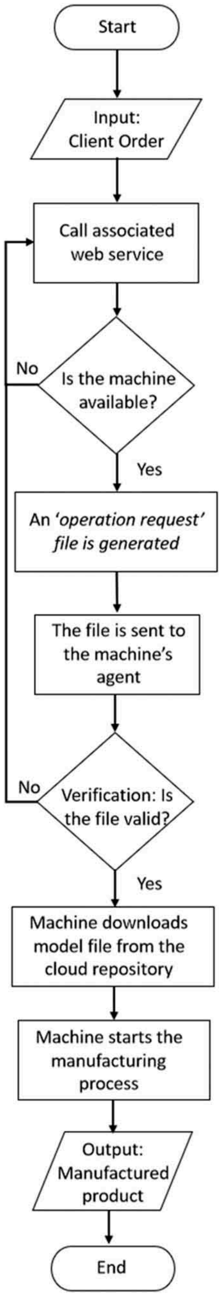


Figure 5. Work flow of the file based operation procedure in the CPMC.

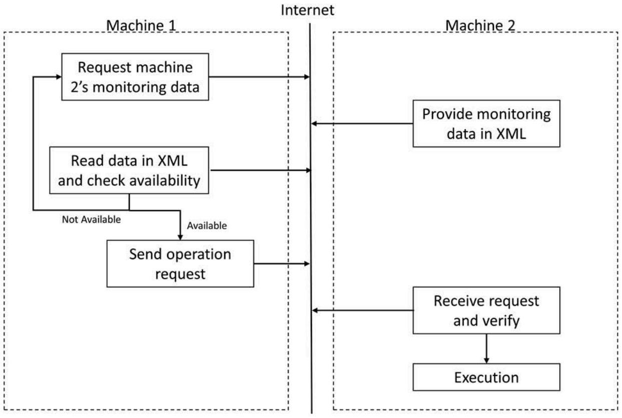


Figure 6. Communication between two machines.

format over the Internet (assuming M_1 knows the address of M_2's agent). If available, M_1's agent generates an *'operation request'* file and sends the file to M_2's agent. M_2's agent then processes the request and completes the operation. If M_2 is not available, M_1's agent has the option of either waiting until M_2 becomes available, or communicating with another machine which is available and can provide the service M_1 is looking for. Figure 6 illustrates the communication process between two machines over the Internet.

The procedure described is very useful in different manufacturing scenarios. It allows manufacturers to achieve factory floor automation, respond to production process failures, perform collaborative manufacturing, improve production scheduling etc. For instance, let there be two identical machines in a factory floor. One machine is assigned a job by the CPMC. During the manufacturing procedure, a failure occurs and the machine becomes incapable of completing the job. At this point, instead of pausing the production process, the agent of machine can forward job to another identical machine in the factory (assuming there is one). If the i.p. address of the second machine's agent is known to the first machine's agent, it generates a new '*operation request'* file for the next machine and forwards the file. The second machine the carries out the job on behalf of the first machine. This is one example scenario, the prospect of such communication between manufacturing resources over the Internet is huge and varied.

One of the key advantages of using the presented communication method between machine tools is the fact that it enables interoperability not only among machines in a single factory floor, but also among machines situated in different locations. At present, in most cases the communication between machines is limited to machines that are situated in close proximity or in a certain factory which are connected with each other through a Local Area Network (LAN). But in the presented communication method, as the machines can communicate over the Internet, they are able to connect to other machines from other factories, from other cities, even from other countries. All machines which are registered and connected to the cloud can access other machines in the CPMC. The '*operation request*' file is generated by the CPMC applications which ensures that only eligible devices' addresses are given in the file. This keeps machines from communicating outside the boundary of the CPMC.

7. Collaborative manufacturing in the CPMC

The capability of communication between machines in the CPMC enables them to participate in collaborative manufacturing actively. They can carry out parts of a single manufacturing job or perform multiple iterations of the same job. The collaborative manufacturing process using the presented communication method of the CPMC minimises the time delay for transferring job from one machine to another as the machines can directly communicate with each other over the Internet.

Like other manufacturing operations discussed above, a collaborative manufacturing job is initiated from a corresponding application in the CPMC. A collaborative manufacturing cloud application is capable of calling web services for all associated machines. Assuming there is n number of machines required for the job, the application contains a list of the addresses of n eligible machines. When there is a request for collaborative job, the collaborative application gathers and stores all the files needed for the job in the cloud's repository and creates links for those files. Then it generates the '*operation request*' file containing all addresses, individual operation types, model file types and links to the model files arranged in sequential order. After that, this file is sent to the first machine's (M_1) agent. When the agent of M_1 sees that the file contains multiple lines, it recognises that this is a collaborative job and it only needs the information on the first line. The agent reads the first line, verifies the information, downloads its own model file and proceeds to execution. When it completes its job, it erases the first line from the '*operation request*' file and reads the address of M_2's agent from the first line (previously second line). Then M_1's agent goes online and checks M_2's current status using the address. If M_2 is available, then M_1's agent sends the '*operation request*' file to M_2's agent and fulfils its responsibility. If M_2 is not available, M_1's agent sends the file back to the cloud. The cloud application then acknowledges the failure and looks for another suitable candidate to replace M_2. When found, the file is delivered to the new machine and the process continues.

The rest of the procedure is similar. M_2 completes its part of the job, deletes the line designated for itself from the file, checks the status of the next machine and forwards it to M_3. The sequence continues until the file reaches M_n machine. When M_n completes its part, it sees that there are no more lines in the file meaning the end of the collaborative job. Figure 7 demonstrates the complete flow of a collaborative manufacturing job with n machines. The machine itself, its adapter and agent are collectively shown as a machine entity.

Let a CPMC system be consists of n identical machines of which each one can complete a certain task w in time period t. Also let a collaborative job require all machines to complete task w once. Here it is assumed that the time between transferring job from one machine to another is negligible, i.e. one

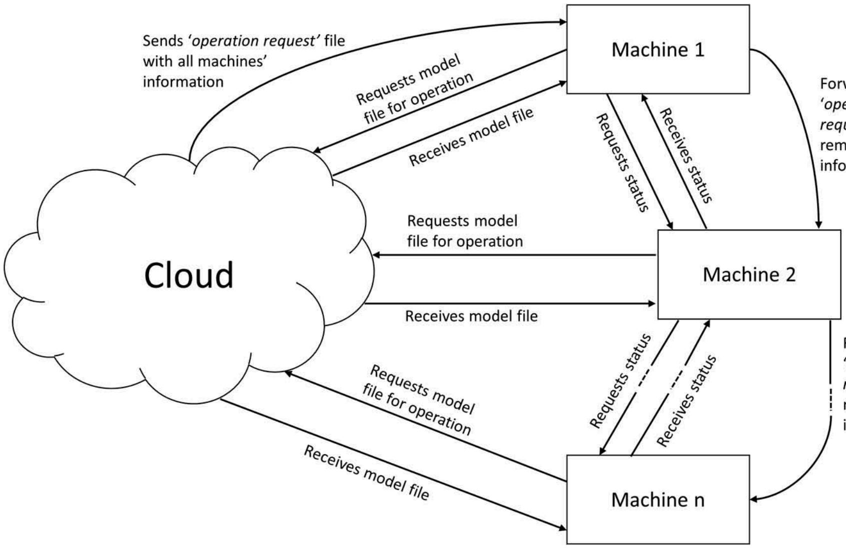
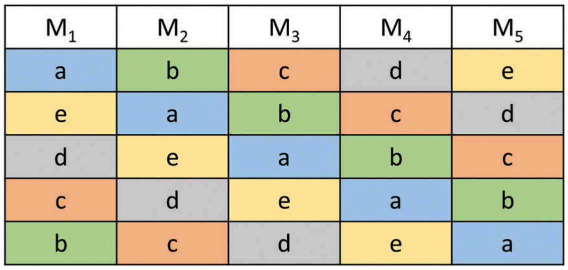


Figure 7. Collaborative manufacturing process in the CPMC.



M_1	M_2	M_3	M_4	M_5
a	b	c	d	e
e	a	b	c	d
d	e	a	b	c
c	d	e	a	b
b	c	d	e	a

Figure 8. Sample case of assigning five consecutive collaborative jobs (a, b, c, d, e) during five time cycles in the CPMC.

machine immediately starts its part as soon as the former machine completes its task. For CPMC, the total time for one such collaborative job is *nt*.

Now let another scenario be considered where *m* collaborative jobs are assigned at the same time. For the CPMC, all the jobs can be initiated at once assuming the initial setup time required for the cloud application to generate the '*operation request*' file is negligible and all machines are primarily available. As all machines are similar and conduct the same task, each can be assigned to a new job. After the first *t* cycle, each machine becomes available and can start the next parts of all the jobs. The situation is illustrated in Figure 8 for $n = 5$ and $m = 5$. So the total time required to complete all jobs is *nt*.

8. Implementation and testbed

To evaluate the performance and feasibility of the CPMC, a fully operational testbed of the CPMC is developed. A

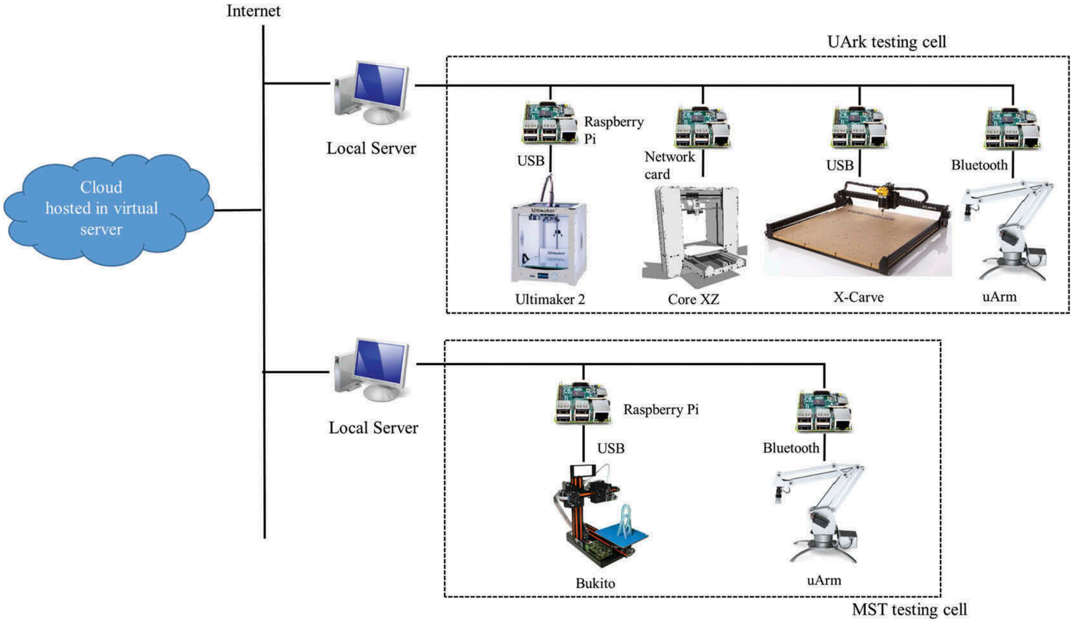


Figure 9. Testbed of the CPMC.

structure of the testbed in presented in Figure 9. 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 (UArk) and the other is in the Missouri University of Science and Technology (MST). The site in UArk has four manufacturing machines, including an X-Carve CNC machine from Inventables, a small robotic arm named uArm from UFactory, a RepRap (Bowyer 2007; Jones et al. 2011) 3D printer Ultimaker 2 from Ultimaker and a custom-made RepRap 3D printer called Core XZ, and three controllers, and one local server. The site in MST has a uArm robotic arm, a RepRap 3D printer Bukito from Bukobot, two controllers, and one local server. All machines used are open-source. Machining tools are connected to its own controller and all the controllers of a site are connected to a local server where the virtualised manufacturing web services are hosted. The local server is a communication focal point from the cloud. The testbed of CPMC is implemented using the presented communication method.

A Raspberry Pi (RPi) works as a controller where the adapter and agent programs of corresponding machine are deployed. Using a RPi offers several advantages. One advantage is its low cost, small size and low power consumption rate. The newest RPis contain enough memory space to provide adequate buffer storage needed by the agent to hold data. It has enough computation power to run multiple agent and adapter programs for several machines simultaneously. For simplicity one RPi has been used for one machine in the testbed. Besides it provides a unique scalable and plug-n-play feature to the system. A new machine can be added to the CPMC system just by connecting it to an RPi where the adapter and agent for that machine is deployed. The RPi also supports various types of standard communication interface such as USB, Serial I/O, Ethernet, Bluetooth and Wifi. In the testbed, the robotic arms are connected to the RPi via Bluetooth, the Core XZ is connected through a network card and the rest are connected via USB.

Both agent and adapter programs of a manufacturing machine run on an RPi. Although one agent can support multiple machines, in this implementation each machine has its own agent and adapter. Both agent and adapter programs are developed in Python. The procedure of acquiring data from the machine varies with machine type. For example, the 3D printers and CNC machines understand G-code, therefore their adapters use G-code commands to fetch information from the machines. For 3D printers two types of files are supported – .gcode/.gco and .stl. In case of the STL model files, the adapter first uses '*slic3r*' (Hodgson 2015) to slice through the file and convert it into a G-code file. For the uArm and the CNC machine, the supported file formats are .csv and G-code files, respectively.

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 centre and several other cloud-based applications. The application centre, which is a web application hosted in the cloud, is a marketplace of all the published applications. Users can sign into the application centre and subscribe cloud-based applications. A dashboard contains access links of the subscribed applications of users. To facilitate publishing web services by the manufacturers, a web-based application named Web Service-Publishing Centre is developed. Manufacturers can publish their web services and also develop their own application. The published web services then appear on the list of services as shown in the Service-Publishing Centre search web page. Users can search for web services by functionalities and other search criteria. While placing an order, the users can choose from available options or parameters. For instance, users can select the colour and material type of a 3D printed object. Cloud applications make HTTP calls to the monitoring web services to acquire data in XML format.

9. Empirical evaluation

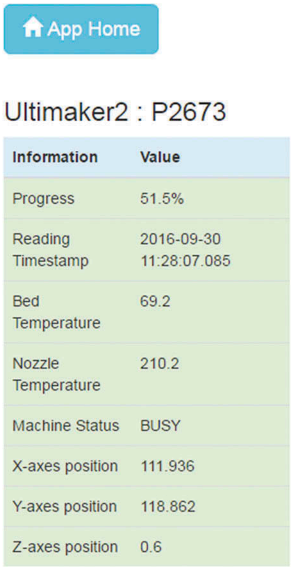
This section describes several manufacturing experiments which have been conducted with the testbed of the CPMC. These test cases have been simulated in order to demonstrate the capability and performance of the presented communication method in different manufacturing scenarios.

9.1. Case 1 – monitoring and operating individual manufacturing machines in CPMC

In this scenario, individual machines have been monitored and assigned manufacturing operations from the cloud applications. Several iterations of the test case have been performed with different conditions. For instance, in one iteration only the Ultimaker 2 3D printer was assigned a job, while other machines were idle. In another iteration all machines of the two testing cells were allotted individual jobs at the same time. Each iteration has been completed successfully. The cloud applications have handled all manufacturing job requests and the machines on both sites completed their assigned process without any failure. Alongside file based jobs, each machine has also been assigned some non-file-based operations. All machines can perform 'STOP' and 'RESET' type operations. Additionally, the CNC machine has 'PAUSE' and 'RESUME' functionality. The robotic arms can remove an object from a pre-specified location to another through the 'REMOVE' operation. All these operations have been tested multiple times. During the operations, the status data of the machines have been displayed accurately in tabular format. Figure 10 shows an example of the monitoring data of the Ultimaker 2 presented by its cloud application during a typical 3D printing operation.

9.2. Case 2 – collaborative manufacturing in CPMC

Several collaborative manufacturing scenarios have also been simulated. In the UArk testing cell, two types of collaborative manufacturing have been implemented. The first one requires the participation of one 3D printer, the robotic arm and the CNC machine. In this scenario, the 3D printer produces an object according to the model file given by the cloud application. When it finishes the production, it requests the robotic arm sitting next to it to remove the object from its heatbed by



Ultimaker2 : P2673

Information	Value
Progress	51.5%
Reading Timestamp	2016-09-30 11:28:07.085
Bed Temperature	69.2
Nozzle Temperature	210.2
Machine Status	BUSY
X-axes position	111.936
Y-axes position	118.862
Z-axes position	0.6

Figure 10. Example of status monitoring of Ultimaker 2 through the CPMC.

forwarding the ‘operation request’ file to it. The robotic arm then takes the object away from the robotic arm. The next part of the collaboration requires the CNC machine to do a drilling job into the 3D printed object. The robotic arm puts the object into the CNC machine’s platform and forwards the ‘operation request’ file to the CNC machine. The CNC machine then starts a drilling job on the object. When it is finishes, the collaborative job is concluded. If any of the machine is busy, then the former machine keeps observing its status and when it becomes available, proceeds with the job. The described collaborative job has been carried out in two different scenarios – when all machines were available throughout the whole job, i.e. not doing some other individual or collaborative job, and when the CNC machine was doing its own individual job. In the latter scenario, when the CNC machine was able to complete its own task before the 3D printer finished its printing, there was no waiting. Figure 11 demonstrates example of status monitoring during different stages of a collaborative manufacturing in the CPMC.

In the second collaborative manufacturing test case, two 3D printers have worked together. When a job is assigned to one 3D printer and it cannot complete the job because of a failure, then it forwards the job to the other 3D printer. In the experiments of this type of collaborative job, some intentional failure scenarios have been orchestrated, i.e. too much high temperature of the extruder, unavailability of printing material, etc. In each experiment, the second 3D printer successfully completed the assigned job.

10. Discussions and future works

The presented communication method of CPMC provides a strong support to the exchange of manufacturing services over the Internet. Manufacturing resources can be virtualised and added to the CPMC using this communication method. The machines can be monitored and operated remotely over the Internet. The presented method also allows manufacturers to perform collaborative manufacturing. Existing industrial communication methods do not provide such facilities for manufacturing machines over the Internet. Most of the existing industrial automation protocols utilise Ethernet based Local Area Network (LAN) and are not designed for communication of manufacturing services over the Internet. Some manufacturers provide Internet connectivity with their machines, but mostly the ability is application or manufacturer specific. MTConnect only provides monitoring data over the Internet, but lacks the remote operation capability. The presented communication method surpasses these limitations and increases interoperability among manufacturing machines by enabling operation of multiple types of machine tools situated in geographically different locations across the Internet.

The cloud based additive manufacturing services mentioned in Section 2 have several issues. They use the Internet for only in the front, for receiving orders from clients and selling products, neither for automating the manufacturing process nor for connecting multiple machines and accessing them remotely. In most cases, whether the printers are physically connected to the Internet or not is unclear. Some service providers offer human-in-the-loop manufacturing execution systems. For example, 3D hubs offer services using networked 3D printers. Users can upload their own model files, choose which material they want to use, select a nearby 3D printing service, send the file to the chosen printer over the Internet and print the file remotely. But the printers are not directly connected to the Internet; the owners of the printers receive the orders, build the products and make deliveries to corresponding clients. In the cases where the machines are said to be networked, details of connection type and communication protocol are missing. Also how the printing jobs are assigned to appropriate 3D printers is not specified. In some cases, like Ponoko and 3D Hubs, clients choose their service providers. Most of other cloud based manufacturing services focus primarily on the analysis of manufacturing machines data. Instead the presented communication method provides direct access to the manufacturing machines located in geographically different locations over the Internet to acquire data and operate the machines. This improves the productivity and scalability of the manufacturing systems through automation and cloud services. Also this communication method allows communication between machine tools over the Internet and enables collaborative manufacturing involving multiple types of machines directly from cloud applications. These features significantly distinguish the presented communication method from other existing industrial communication protocols and standards. The case studies with the testbed of the CPMC discussed in the previous section demonstrate the effectiveness of the communication method with three types of machines in simplified manufacturing scenarios.

Although the experiments conducted with the testbed using the agent–adapter based communication method show feasibility of integration of manufacturing cloud and cyber physical systems to enable direct operation and monitoring of machining tools from a CPMC over the Internet,

Ultimaker2 : P2673

Information	Value
Progress	2.1%
Reading Timestamp	2016-09-30 11:27:25.085
Bed Temperature	68.0
Nozzle Temperature	209.5
Machine Status	BUSY
X-axes position	105.13
Y-axes position	117.556
Z-axes position	0.3

X-Carve : C153

Information	Value
Progress	0.0%
Reading Timestamp	2016-09-30 11:27:25.103
Machine Status	Alarm
X-axes position	0.000
Y-axes position	0.000
Z-axes position	0.000

Uarm : #1

Information	Value
Reading Timestamp	2016-09-30 11:27:25.116
Machine Status	available
Robot Grabber	2
Robot Grab Rotation	0
X-axes position	0
Y-axes position	0
Z-axes position	0

Ultimaker2 : P2673

Information	Value
Progress	0.0%
Reading Timestamp	2016-09-30 11:29:01.096
Bed Temperature	67.7
Nozzle Temperature	185.8
Machine Status	AVAILABLE
X-axes position	0.0
Y-axes position	0.0
Z-axes position	0.0

X-Carve : C153

Information	Value
Progress	0.0%
Reading Timestamp	2016-09-30 11:29:01.107
Machine Status	Alarm
X-axes position	0.000
Y-axes position	0.000
Z-axes position	0.000

Uarm : #1

Information	Value
Reading Timestamp	2016-09-30 11:29:01.120
Machine Status	BUSY
Robot Grabber	1
Robot Grab Rotation	0
X-axes position	-120
Y-axes position	180
Z-axes position	90

Ultimaker2 : P2673

Information	Value
Progress	0.0%
Reading Timestamp	2016-09-30 11:29:50.093
Bed Temperature	62.5
Nozzle Temperature	123.1
Machine Status	AVAILABLE
X-axes position	0.0
Y-axes position	0.0
Z-axes position	0.0

X-Carve : C153

Information	Value
Progress	48.65%
Reading Timestamp	2016-09-30 11:29:50.106
Machine Status	Run
X-axes position	682.750
Y-axes position	580.000
Z-axes position	-4.191

Uarm : #1

Information	Value
Reading Timestamp	2016-09-30 11:29:50.117
Machine Status	available
Robot Grabber	2
Robot Grab Rotation	0
X-axes position	0
Y-axes position	0
Z-axes position	0

Figure 11. Status monitoring during different stages of a collaborative manufacturing scenario in the CPMC. (a) When Ultimaker 2 is working and other two machines are available. (b) When robotic arm is working and other two machines are available. (c) When CNC machine is working and other two machines are available.

additional researches and experiments on traditional large-scale machine tools in diverse manufacturing environments are required before achieving its full potential in the industry. Collaboration with manufacturing and IT companies to conduct more research is also necessary. 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 and scheduling, dynamic configurability of the manufacturing resources, MESs, and manufacturing data analytics and management.

This research on the agent–adapter based communication method provides valuable insights on the development of an Internet-scale manufacturing communication protocol. MTConnect standard and HTTP based communication are used to connect manufacturing tools over the Internet. However, more improvements are essential to satisfy several requirements of Internet scale manufacturing communication methods for easy and efficient exchange of manufacturing services over the Internet. One of the biggest issues regarding CMfg is cybersecurity. It is very important to ensure the machines safety from malicious attacks from the outside, especially for operating machines over the Internet. More researches and experiments need to be done on the presented communication method to make it a secure Internet scale communication method. Extending the presented method to establish communication between cloud services and other IoT devices alongside manufacturing machines is also promising.

Not all existing industrial machine tools are capable of computation and networking. Besides, many existing machine tools are operated by legacy and outdated operating systems. Although addition of those machines to a CPMC is challenging, several types of those have successfully included the testbed of CPMC using the presented communication method. The machine tools used in the testbed did not have the networking capabilities except the USB interface. The agent and adapter programs deployed in the Raspberry Pi are used to connect them to the network. However, for large industrial machine tools, small Raspberry Pi controllers may not be sufficient. 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.

11. Conclusions

The integration of two emerging manufacturing paradigms – cyber–physical systems and CMfg is promising in transforming manufacturing. It leads to development of cyber-physical manufacturing cloud. In this paper, a communication method has been developed for exchanging manufacturing services over the Internet in the CPMC based on MTConnect and HTTP. Its use of MTConnect standard and HTTP based communication allows the CPMC to monitor and operate connected machining tools across the Internet. Different types of machines can be added to the CPMC easily through adding the controller containing agents and adapters to the machines. The presented communication method of the manufacturing services in the CPMC allows the clients to browse and select their desired products from the cloud. It also provides the manufacturers with the opportunity of offering their services over the Internet with much ease and creating customised on-demand solutions to save costs and improve profits. Clients and manufacturers can also monitor the progress of the orders and status of the machines in the manufacturing process over the Internet. In addition, it enables collaborative manufacturing by allowing direct communication of manufacturing services among machining tools over the Internet. This communication method provides a foundation of the first testbed of a cyber-physical manufacturing cloud to allow direct operations of machining tools over the Internet. The experiments conducted with the testbed demonstrates excellent feasibility and effectiveness of the communication method and cyber–physical manufacturing cloud for manufacturing with machines in geographically distributed locations.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

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

References

Bowyer, A. 2007. "The Self-Replicating Rapid Prototyper-Manufacturing for the Masses." In *8th National Conference on Rapid Design, Prototyping & Manufacturing*. University of Bath. http://opus.bath.ac.uk/1684/

Brooks, P. 2001. "Ethernet/IP-Industrial Protocol." In *Proceedings. 2001 8th IEEE International Conference on Emerging Technologies and Factory Automation*, Vol. 2, 505–514. IEEE.

ControlNet International and open DeviceNet Vendor association. 2001. "Ethernet/IP Adaptation of CIP Specification, Release 1.0." Accessed September 10 2016. http://read.pudn.com/downloads166/ebook/763212/EIP-CIP-V2-1.0.pdf

Drath, R., A. Luder, J. Peschke, and L. Hundt. 2008. "AutomationML-The Glue for Seamless Automation Engineering." In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, 616–623. IEEE. http://ieeexplore.ieee.org/abstract/document/4638461/

Erickson, K. T. 1996. "Programmable Logic Controllers – The Workhorse of Factory Automation Keeps Things on Track." *IEEE Potentials* 15 (1): 14–17. doi:10.1109/45.481370.

Fieldbus Foundation. 2001. "High Speed Ethernet Specification Documents FF-801, 803, 586, 588,589, 593, 941." Accessed September 9 2016. http://www.fieldbus.org.

Hancu, O., V. Maties, R. Balan, and S. Stan. 2007. "Mechatronic Approach for Design and Control of a Hydraulic 3-Dof Parallel Robot." In *Annals of DAAAM & Proceedings*, 321–323. http://go.galegroup.com/ps/i.do?id=GALE%7CA177174624&sid=googleScholar&v=2.1&it=r&linkaccess=fulltext&issn=17269679&p=AONE&sw=w

Hellinger, A., and H. Seeger. 2011. *Cyber-Physical Systems. Driving Force for Innovation in Mobility, Health, Energy and Production*. Acatech Position Paper, National Academy of Science and Engineering: 2. Berlin: Springer

Henssen, R., and M. Schleipen. 2014. "Interoperability between OPC UA and AutomationML." *Procedia CIRP* 25: 297–304. doi:10.1016/j.procir.2014.10.042.

Hodgson, G. 2015. "Slic3r Manual." LulzBot. Accessed September 24 2016. http://manual.slic3r.org/intro/overview

Holtewert, P., R. Wutzke, J. Seidelmann, and T. Bauernhansl. 2013. "Virtual Fort Knox Federative, Secure and Cloud-Based Platform for Manufacturing." *Procedia CIRP* 7: 527–532. doi:10.1016/j.procir.2013.06.027.

IEC (International Electrotechnical Commission). 2004e. "IEC 65C/358/NP. Real-time Ethernet: SERCOS III."

IEC (International Electrotechnical Commission). 2004f. "IEC 65C/356/NP. Real-time Ethernet: POWERLINK."

IEC (International Electrotechnical Commission). 2004g. "IEC 65C/355/NP. Real-time Ethernet: ETHERCAT."

IEC (International Electrotechnical Commission). 2004h. "IEC (2004h). IEC 65C/353/NP. Real-time Ethernet: TCnet (Time-critical Control Network)."

International Electrotechnical Commission (IEC). 2004a. "IEC 65C/359/NP. Real-time Ethernet: PROFINET IO. Application Layer Service Definition & Application Layer Protocol Specification."

International Electrotechnical Commission (IEC). 2004c. "IEC 65C/361/NP. Real-time Ethernet: EtherNet/IP with Time Synchronization."

Jones, R., P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, and A. Bowyer. 2011. "RepRap–The Replicating Rapid Prototyper." *Robotica* 29 (01): 177–191. doi:10.1017/S026357471000069X.

Kagermann, H., J. Helbig, A. Hellinger, and W. Wahlster. 2013. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the INDUSTRIE 4.0 Working Group*. Forschungsunion. http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf

Khaitan, S. K., and J. D. McCalley. 2015. "Design Techniques and Applications of Cyberphysical Systems: A Survey." *IEEE Systems Journal* 9 (2): 350–365. doi:10.1109/JSYST.2014.2322503.

Labs, U. I. 2017. "O3 – Operate, Orchestrate, and Originate – 14-06-05." Accessed July 12 2017. http://www.uilabs.org/project/o3-operate-orchestrate-and-originate-14-06-05/

Leitner, S. H., and W. Mahnke. 2006. *OPC UA – Service-Oriented Architecture for Industrial Applications*. ABB Corporate Research Center. http://www2.cs.uni-paderborn.de/cs/ag-engels/GI/ORA2006-Papers/leitner-final.pdf

Li, B. H., L. Zhang, S. L. Wang, F. Tao, J. W. Cao, X. D. Jiang, X. Song, and X. D. Chai. 2010. "Cloud Manufacturing: A New Service-Oriented Networked Manufacturing Model." *Computer Integrated Manufacturing Systems* 16 (1): 1–7.

Li, R. F., Q. Liu, and W. J. Xu. 2012. "Perception and Access Adaptation of Equipment Resources in Cloud Manufacturing." *Computer Integrated Manufacturing Systems* 18 (7): 1547–1553.

Lin, Y. L., C. C. Lin, and H. S. Chiu. 2015. "The Development of Intelligent Service System for Machine Tool Industry." In *2015 1st International Conference on Industrial Networks and Intelligent Systems (Iniscom)*, 100–106. IEEE. http://ieeexplore.ieee.org/abstract/document/7157829

Liu, N., X. Li, and Q. Wang. 2011. "A Resource & Capability Virtualization Method for Cloud Manufacturing Systems." In *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1003–1008. IEEE. http://ieeexplore.ieee.org/document/6083800/

Liu, X. F., M. R. Shahriar, S. M. N. A. Sunny, M. C. Leu, M. Cheng, and L. Hu. 2016. "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 (IDETC/CIE)*, V01BT02A019. ASME. http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=2591530

Liu, X. F., M. R. Shahriar, S. M. N. A. Sunny, M. C. Leu, M. Cheng, and L. Hu. 2017. "Cyber-Physical Manufacturing Cloud: Architecture, Virtualization, Communication, and Testbed." *Journal of Manufacturing Systems* 43 (2): 352–364. doi:10.1016/j.jmsy.2017.04.004.

Liu, X. F., S. M. N. A. Sunny, M. R. Shahriar, M. C. Leu, M. Cheng, and L. Hu. 2016. "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 (IDETC/CIE)*, V01AT02A019. ASME. http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=2591461

Luo, Y. L., L. Zhang, D. J. He, L. Ren, and F. Tao. 2011. "Study on Multi-View Model for Cloud Manufacturing." In *Advanced Materials Research*. Vol. 201, 685–688. Trans Tech Publications. https://www.scientific.net/AMR.201-203.685

Mai, J., L. Zhang, F. Tao, and L. Ren. 2016. "Customized Production Based on Distributed 3D Printing Services in Cloud Manufacturing." *The International Journal of Advanced Manufacturing Technology* 84 (1–4): 71–83. doi:10.1007/s00170-015-7871-y.

Modern Applications News. 2016. "M2M Communications can Improve Shop's Output Up to 50% and Increase Income up to 6 Times". Accesses September 15 2016. http://www.modernapplicationsnews.com/cms/man/opens/article-view-man.php?nid=2&bid=374&et=featurearticle&pn=02

McFarlane, A. 1997. "Fieldbus Review." *Sensor Review* 17 (3): 204–210. doi:10.1108/02602289710172319.

Michaloski, J., B. Lee, F. Proctor, S. Venkatesh, and S. Ly. 2009. "Quantifying the Performance of MTConnect in a Distributed Manufacturing Environment." In *Proceddings of ASME DETC Conference*, 533–539. ASME. http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1649239

Modbus Organization. 2006. "Modbus Application Protocol Specification". Accessed September 18 2016. http://www.modbus.org/docs/Modbus_Application_protocol_V1_1b.pdf.

Monostori, L. 2014. "Cyber-Physical Production Systems: Roots, Expectations and R&D Challenges." *Procedia CIRP* 17: 9–13. doi:10.1016/j.procir.2014.03.115.

Monostori, L., B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda. 2016. "Cyber-Physical Systems in Manufacturing." *CIRP Annals-Manufacturing Technology* 65 (2): 621–641. doi:10.1016/j.cirp.2016.06.005.

MTConnect Institute. 2016. "MTConnect Overview & Architecture". Accessed September 20 2016. http://www.mtconnect.org/resources/

Neil, S. 2016. "MES Meets the Cloud." *Automation World*, May 19. https://www.automationworld.com/article/industry-type/discrete-manufacturing/mes-meets-cloud

Neumann, P. 2007. "Communication in Industrial Automation — What Is Going On?." *Control Engineering Practice* 15 (11): 1332–1347. doi:10.1016/j.conengprac.2006.10.004.

NSF (US National Science Foundation) 2011, "Cyber-Physical Systems (CPS)." July 6. Accessed September 22 2016. https://www.nsf.gov/pubs/2011/nsf11516/nsf11516.pdf

Rad, C.-R., O. Hancu, I.-A. Takacs, and G. Olteanu. 2015. "Smart Monitoring of Potato Crop: A Cyber-Physical System Architecture Model in the Field of Precision Agriculture." *Agriculture and Agricultural Science Procedia* 6: 73–79. doi:10.1016/j.aaspro.2015.08.041.

Rauschecker, U., M. Meier, R. Muckenhirn, A. L. K. Yip, A. P. Jagadeesan, and J. Corney. 2011. *Cloud-Based Manufacturing-As-A-Service Environment for Customized Products*. International Information Management Corporation Limited: eChallenges e-2011. https://strathprints.strath.ac.uk/38573/

Rayna, T., L. Striukova, and J. Darlington. 2015. "Co-Creation and User Innovation: The Role of Online 3D Printing Platforms." *Journal of Engineering and Technology Management* 37: 90–102. doi:10.1016/j.jengtecman.2015.07.002.

ROY-G-BIV. 2017. "ROY-G-BIV | Keep it Moving." Accessed July 12 2017. http://www.roygbiv.com

Sauter, T. 2010. "The Three Generations of Field-Level Networks—Evolution and Compatibility Issues." *IEEE Transactions on Industrial Electronics* 57 (11): 3585–3595. doi:10.1109/TIE.2010.2062473.

Siemens. 2017. "MindSphere." Accessed July 12 2017. https://www.siemens.com/global/en/home/products/software/mindsphere.html

Swales, A. 1999. "Open Modbus/TCP Specification." *Schneider Electric* 29. http://www.dankohn.info/projects/Fieldpoint_module/Open_ModbusTCP_Standard.pdf

Tao, F., L. Zhang, V. C. Venkatesh, Y. Luo, and Y. Cheng. 2011. "Cloud Manufacturing: A Computing and Service-Oriented Manufacturing Model." *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 225: 1969–1976. doi:10.1177/0954405411405575.

Tao, F., Y. Cheng, L. Da Xu, L. Zhang, and B. H. Li. 2014. "CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System." *IEEE Transactions on Industrial Informatics* 10 (2): 1435–1442. doi:10.1109/TII.2014.2306383.

Tao, F., Y. Zuo, L. Da Xu, and L. Zhang. 2014. "IoT-based Intelligent Perception and Access of Manufacturing Resource toward Cloud Manufacturing." *IEEE Transactions on Industrial Informatics* 10 (2): 1547–1557. doi:10.1109/TII.2014.2306397.

ThomasNet. 2010. "OPC Foundation and MTConnect Institute Announce a Memorandum of Understanding." Accessed September 15 2016. http://news.thomasnet.com/companystory/memorandum-of-understanding-announced-between-opc-mtconnect-584167

Thomesse, J.-P. 2005. "Fieldbus Technology in Industrial Automation." *Proceedings of the IEEE* 93 (6): 1073–1101. doi:10.1109/JPROC.2005.849724.

Tindell, K. W., H. Hansson, and A. J. Wellings. 1994. "Analysing Real-Time Communications: Controller Area Network (CAN)." In *Real-Time Systems Symposium, 1994, Proceedings*, 259–263. https://pdfs.semanticscholar.org/ac88/f1ea5db2fd355e458f08227b15bec910a8ec.pdf.

Vijayaraghavan, A., W. Sobel, A. Fox, D. Dornfeld, and P. Warndorf. 2008. "Improving Machine Tool Interoperability Using Standardized Interface Protocols: MT Connect." *Laboratory for Manufacturing and Sustainability*. https://escholarship.org/uc/item/4zs976kx

Vincent, S. J. 2001. "FOUNDATION™ Fieldbus High Speed Ethernet Control System." Accessed September 9 2016. http://www.fieldbusinc.com/downloads/hsepaper.pdf

Wang, L. 2013. "Machine Availability Monitoring and Machining Process Planning Towards Cloud Manufacturing." *CIRP Journal of Manufacturing Science and Technology* 6 (4): 263–273. doi:10.1016/j.cirpj.2013.07.001.

Wang, L., R. Gao, and I. Ragai. 2014. "An Integrated Cyber-Physical System for Cloud Manufacturing." In *International Manufacturing Science and Engineering Conference, Volume 1: Materials; Micro and Nano Technologies; Properties, Applications and Systems; Sustainable Manufacturing*, V001T04A029. ASME. http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1913987

Wikipedia. 2014. "CC-Link Industrial Networks". Last modified August 15. Accessed September 9 2016. http://en.wikipedia.org/wiki/CC-Link_Industrial_Networks/

Wikipedia. 2017. "Predix (Software)." Last modified June 14. Accessed July 12 2017. http://en.wikipedia.org/wiki/Predix_(software)

Wu, D., D. W. Rosen, L. Wang, and D. Schaefer. 2015. "Cloud-Based Design and Manufacturing: A New Paradigm in Digital Manufacturing and Design Innovation." *Computer-Aided Design* 59: 1–14. doi:10.1016/j.cad.2014.07.006.

Wu, D., J. L. Thames, D. W. Rosen, and D. Schaefer. 2012. "Towards a Cloud-Based Design and Manufacturing Paradigm: Looking Backward, Looking Forward." In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 315–328. http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1736119

Wu, D., M. J. Greer, D. W. Rosen, and D. Schaefer. 2013. "Cloud Manufacturing: Strategic Vision and State-Of-The-Art." *Journal of Manufacturing Systems* 32 (4): 564–579. doi:10.1016/j.jmsy.2013.04.008.

Xiang, F., and Y. F. Hu. 2012. "Cloud Manufacturing Resource Access System Based on Internet of Things." In *Applied Mechanics and Materials*. Vol. 121, 2421–2425. Trans Tech Publications. https://www.scientific.net/AMM.121-126.2421

Xu, X. 2012. "From Cloud Computing to Cloud Manufacturing." *Robotics and Computer-Integrated Manufacturing* 28 (1): 75–86. doi:10.1016/j.rcim.2011.07.002.

Zuehlke, D. 2010. "SmartFactory—Towards a Factory-of-Things." *Annual Reviews in Control* 34 (1): 129–138. doi:10.1016/j.arcontrol.2010.02.008.

Zurawski, R., ed. 2014. *Industrial Communication Technology Handbook*. CRC Press. Spain. https://books.google.com/books?id=ppzNBQAAQBAJ&printsec=frontcover&dq=Industrial+Communication+Technology+Handbook&hl=en&sa=X&ved=0ahUKEwixq_SapdDXAhWg0YMKHTCpDTMQ6AEIKzAB#v=onepage&q=Industrial%20Communication%20Technology%20Handbook&f=false