Smart Manufacturing Systems based on Cyber-physical Manufacturing Services (CPMS)

Yan Lu* and Feng Ju**

*National Institute of Standards and Technology, Gaithersburg, MD 20899
USA (Tel: 301-975-8228; e-mail: yan.lu@nist.gov)
**School of Computing, Informatics, and Decision Systems Engineering, Arizona State University,
Tempe, AZ 85281 USA (Tel: 480-965-1746; e-mail: fengju@asu.edu)

Abstract:

Future manufacturing is becoming "smart" – capable of agilely adapting to a wide variety of changing conditions. This requires production plants, supply chains and logistic systems to be flexible in design and reconfigurable "on the fly" to respond quickly to customer needs, production uncertainties, and market changes. Service-Oriented Architecture (SOA) provides a promising platform to achieve such manufacturing agility. It has proven effective for business process adaptation. When combined with the emerging Internet of Things (IoT) technology and the concept of cyber-physical production systems, it is expected to similarly revolutionize real-time manufacturing systems. This paper proposes a new concept of cyber-physical manufacturing services (CPMS) for service-oriented smart manufacturing systems. In addition, we propose a modeling framework that provides appropriate conceptual models for developing and describing CPMS and enabling their composition. Specifically, the modeling framework separates service provision models from service request models and proposes the use of standardized functional taxonomies and a reference ontology to facilitate the mediation between service requests and service consumptions. A 3D-printing use case serves as an example implementation of an SOA-based smart manufacturing system based on our proposed modeling framework.

Keywords: Service-oriented architecture, cyber-physical manufacturing service, real-time, smart manufacturing system, 3D printing, additive manufacturing

---

1. INTRODUCTION

Historically, manufacturing systems have been designed along relatively rigid hierarchical architectures, the scopes of which are limited to the arrangement and operation of resources directly related to production. The ISA95 model[1] classifies manufacturing functions into five logically separated layers. In a typical manufacturing system, lower level systems (0, 1, and 2) are called Operational Technology (OT)[2] functions, and include hardware and software that monitor and control physical processes. Higher-level functions (3 and 4) are considered Information Technology (IT) domain functions covering manufacturing operations management (MOM), enterprise resource planning (ERP), and more. Because manufacturing IT and OT applications are managed by different organizations within enterprises, their integration is difficult. As a result, traditional manufacturing systems are slow in responding to market or supply chain changes and are very vulnerable to malfunctions of main subsystems. Furthermore, the rigid architecture of traditional manufacturing systems also leads to the difficulty of reusing or retrofitting existing manufacturing capability. Manufacturers tend to replace old assets with new, incurring huge capital investment costs and expending considerable engineering effort.

In contrast, the emerging trend of smart manufacturing (SM) transforms manufacturing systems into agile and efficient ecosystems. SM integrates the latest information and communication technologies (ICT) into manufacturing systems to enable real-time response to changing demands and conditions in the factory, in the supply network, and in customer needs. In this new paradigm, the Internet of Things (IoT), the digital factory, and cloud computing technology play major roles in transforming the rigid hierarchical architecture into a flexible style (Kulvatunyou, (2016)). Lu et al. (2016b) propose a Service Oriented Architecture (SOA)-based approach to restructure manufacturing systems and make them smarter and more agile. SM systems based on SOA adopt modular design of both hardware and software and combine these elements into cyber-physical systems (CPS) that can then be made available as services for easy access and consistent reuse.

In analogy to the software-based computing services common in the IT world, CPS services are common units for SOA-based SM systems. In our research, we refer to cyber-physical

---

[1] https://isa-95.com/

[2] http://www.gartner.com/it-glossary/operational-technology-ot/

manufacturing services (CPMS). The scope of CPMS covers both processing units in factories and logistic components in supply chains. The criticality of matching physical location, scheduling availability, and capability of a CPMS to job requirements in real time make the design of SOA-based smart manufacturing systems quite challenging.

This paper extends an SOA approach for integrating business applications to smart manufacturing system design based on the concept of CPMS. The main contribution of the paper is the description of the information models and methods that would enable easy development, usage, and dynamic composition of manufacturing and logistics services. We first present a literature review on the application of SOA to manufacturing systems. Then we introduce a definition of CPMS. Section 4 presents a CPMS modeling framework covering the design and application perspectives of CPMS-based manufacturing systems. In Section 5, we use a case study of an SOA-based real-time shop floor manufacturing system to illustrate the application of the CPMS modeling framework. Section 6 concludes the paper and discusses future work.

2. BACKGROUND

An SOA defines a collection of principles for distributed systems design. It relies on the integration of different system components that provide functionalities as loosely-coupled services over a network. These functionalities are packaged via standardized interfaces independent from the underlying implementation and location. SOA has been widely used to achieve adaptable business processes since 1980s (Krafzig et al. (2004)).

Web service technologies enable the most prevailing implementation of SOA[3]. XML (Extensible Markup Language)-based languages like WSDL (Web Service Description Language), OWL-S (Semantic Markup for Web Services). are used to describe a service's interface, functionalities, and characteristics. The UDDI (Universal Description Discovery and Integration) specification depicts a way to build service repositories and specifies how services can be published by service providers and searched and discovered by service requesters. Composition of existing web services into more complex services can be done using BPEL (Business Process Execution Language). The services can be invoked based on the interface description in WSDL. SOAP (Simple Object Access Protocol) messages are used for such invocations. Besides standards-based web service technologies, ESB (Enterprise Service Bus) is also considered as a de facto method for SOA[4].

In the manufacturing domain, the concept of SOA has been practically applied in enterprise level integration. At the shop-floor level, quite a few research projects have applied SOA concepts to improve manufacturing system flexibility and reconfigurability[5, 6]. For example, agent-based distributed intelligent manufacturing systems and holonic manufacturing systems – were heavily studied in academia and industry before the early 2000s (Marik (2003)). However, agent-based or holonic manufacturing systems do not exercise some capabilities of SOA systems. They don't specifically address composability issues and the standards necessary to integrate heterogonous agent systems did not exist. In the last several years, the SOA approach to automation was studied and developed with the aim of providing methods, models, and technologies to reduce automation engineering (Theorin et al. (2012, 2014), Vyatkin et al. (2014)). Other influential projects applying SOA to industrial automation include SIRENA[7]and SCORDES[8], each focusing on developing a web-service-based communication architecture for field devices integration. The projects resulted in Device Profile for Web Service standards[9] and their solution stacks for industrial automation. Separate efforts to apply OPC UA (OLE for Process Control Unified Architecture)[10] to shop floor integration also promote the SOA approach, such as the one demonstrated by SmartFactoryFL[11]. Overall, these efforts took bottom-up approaches and generated results for specific domains (e.g., automation engineering). There is no generic framework to enable SOA methods to integrate functions of the entire manufacturing ecosystems including OT, IT, and supply chain logistic systems.

Automatic service discovery, identification, and orchestration (choreography) to accomplish tasks are very important capabilities for SOA adoption. In the web service area, tremendous effort has been put toward automatic web service integration based on semantics. Although the concepts of semantic web and semantic web services haven't achieved their vision since those concepts cannot scale to meet the general web demands (Hendler, (2011)), some of the results from earlier research projects are very insightful for a domain-specific implementation. For example, the Web Service Modeling Framework (WSMF) (Fensel et al, (2002)) proposed a modeling framework describing various web-service aspects to support strong decoupling of the system components and strong mediation of service integration. Figure 1 shows the core elements of the modeling framework.

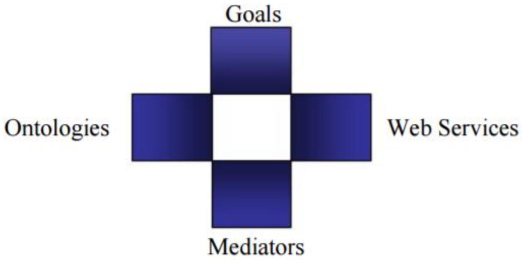


Fig. 1. WSMF core elements for web service integration (Domingue et al. (2005))

As shown in Figure 1, the WSMF consists of four core elements: goal repositories define service requests; web services descriptions define various aspects of a web service; ontolo-

---

[3] https://www.w3.org/TR/ws-arch/

[4] https://en.wikipedia.org/wiki/Enterprise_service_bus

[5] https:/ims.org

[6] https://cordis.europa.eu/esprit/src/22728.htm

[7] http://www.ws4d.org/projects/sirena/

[8] http://www.socrades.net/

[9] http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.html

[10] https://opcfoundation.org/about/opc-technologies/opc-ua/

[11] http://www.smartfactory-kl.de/

gies provide the terminology domain knowledge; and mediators do semantic matching between service requests and services provisions and facilitate the invocation.

Following the semantic web approach, there have been several ontologies developed to formally model manufacturing systems and to enhance sharing and reuse of manufacturing knowledge. Examples of these ontologies include P2 Ontology (Diep et al. (2007)), PSL[12], MASON (Lemaignan et al. (2006)). PSL was adopted by ISO as an international standard in ISO 18629.

Despite these efforts, manufacturing services based on CPS haven't yet been studied as far as practical integration of OT, IT, and supply chain logistic systems. Our work aims to provide methods, standards, and tools to develop and integrate CPS-based manufacturing services within the scope of a whole manufacturing ecosystem (Lu et al. (2016a)).

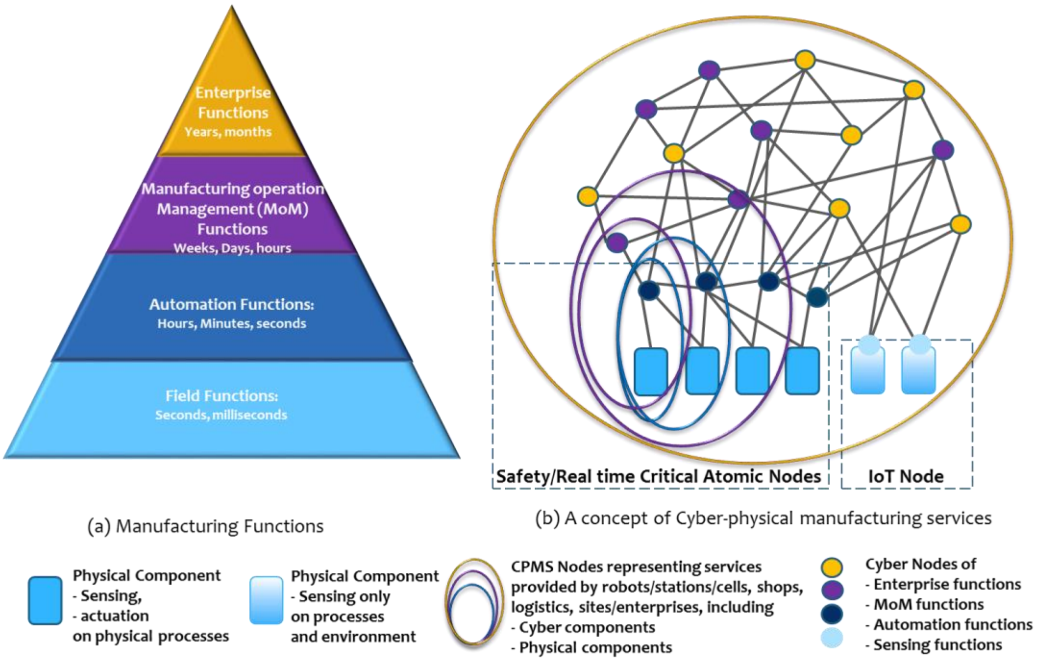
3. CPMS AND SOA-BASED SMART MANUFACTURING

The fusion of ICT technologies changes both the scope of manufacturing functions and the interactions among them. While IoT enables unified communications among manufacturing system components, cloud and mobile computing enable manufacturing functions once implemented at different levels of the hierarchy to now be available without even knowing where they are executed. Smart components and smart systems on the shop floor can run advanced analytics and simulations, and make decisions beyond the lower functions defined in ISA 95. To realize the vision of smart manufacturing where systems respond in real time to changing demands and conditions in the factory, in the supply network, and in customer needs, the classical manufacturing system architectural paradigm based on a hierarchical control model (Figure 2 (a)) must evolve. A new paradigm based on distributed manufacturing services is starting to be adopted, as shown in Figure 2 (b).

In Figure 2 (b), SOA-based smart manufacturing systems are composed of loosely coupled service nodes. Manufacturing services are classified as two types: Cyber (computing) Service and Cyber-physical Manufacturing Services (CPMS).

Traditional software-based IT domain services are classified as Cyber Service nodes. These include enterprise functions such as ERP, Supply Chain Management, MOM functions, and engineering functions associated with Product Lifecycle Management. In addition, virtual factory software for modeling and simulation, data analytics and visualization tools, and other IoT nodes are also regarded as Cyber Service nodes in smart manufacturing systems. Integration of purely Cyber Service nodes is similar to today's SOA approach.

The other type of smart manufacturing service node is named CPMS and differentiates the nature of smart manufacturing systems from existing SOA systems. The formation of CPMS partly breaks the traditional hierarchical automation pyramid and turns the manufacturing ecosystem into a distributed architecture. Depicted as circles in Figure 2 (b), CPMS nodes can represent services provided by work stations, conveyor belts, robots, cells, shops, plants, factories and entire enterprises. To leverage safety and security requirements, level 0 and 1 components of the ISA model, which include real-time and safety critical functions, are encapsulated inside atomic services and stay untouched in this architecture. The atomic CPMS nodes and newly added IoT devices can be nested and composed upwards as CPMS that will be available for shop-floor-level distributed manufacturing, decisions at the MOM and enterprise levels, and even for global production planning



(a) Manufacturing Functions

(b) A concept of Cyber-physical manufacturing services

Fig.2. Concept of Cyber-Physical Manufacturing Services

---

[12] http://www.mel.nist.gov/psl/psl-ontology/psl_core.html

in a connected world. The real-time requirements of the composite services have to be maintained through the composition process. We define CPMS as follows:

CPMS are manufacturing services that include both cyber components with computing and network capabilities and physical components. In addition, the physical components should be able to either process material, store partially finished products, or transport materials and parts.

With the definition of CPMS, our proposed SOA-based SMS has additional requirements for research and implementation that can't be satisfied by existing frameworks. A modeling framework is proposed to help develop such systems. The details will be illustrated in the next section.

4. MODELING FRAMEWORK FOR CYBER-PHYSICAL MANUFACTURING SERVICES

Implementing the CPMS based on SOA is neither simple, trivial, nor quick. It might take many years to fully realize the strength of the new architecture. Specifically, several aspects are considered for designing successful CPS-based services for SMS to cover the complete spectrum of service representation, discovery, understanding/validation, negotiation, invocation, monitoring, fault handling, and service adaptation. In addition, for dynamic and automatic service composition and execution, one must decide:

1) What is the best functional decomposition of CPMS for maximal manufacturing system adaptability?
2) How to enable automated orchestration possible to enable dynamic reconfiguration?
3) How to assure non-functional requirements are met?
4) How to operate and evolve the services once deployed?

To address these challenges, a modeling framework is necessary for designing, developing, and operating CPMS-based SMS. The CPMS model framework should cover the modeling perspectives for automated composition. Specifically, we need to consider constraints based on the physical component (context-based) and the services it provides. In addition, to satisfy the needs for real-time decision making, the states of physical system-based services should be available for query. A methd defining state models for CPMS is needed. Manufacturing goals and decomposition are critical as well in correlating manufacturing activities with services.

Figure 3 depicts a CPMS-based service-modeling framework that extends the WSMF framework. Semantic technology is adopted to separate service consumers in a specific context from the services to be consumed. In consequence, we model the service consumption side and service provision side separately. Domain-specific and even enterprise-specific models are accepted by the modeling framework. To match the manufacturing service goals with services, we employ a manufacturing reference ontology that can be used by manufacturing service mediators for service registration, discovery, match, identification, optimization, orchestration, and invocation.

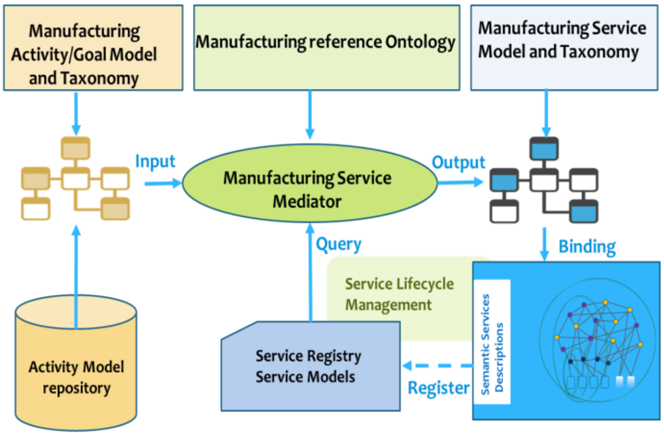


Fig. 3. Modelling framework for Cyber-Physical Manufacturing Services

4.1 *CPMS Service Modeling*

CPMS models should consider explicit capability representation, standard interfaces, grounding mechanisms, and non-functional features. The Dublin Core[13] Metadata Element Set can be used to capture general service property sets. The granularity and type of CPMS can be defined and standardized for individual domains. CPMS capability is the key to service matching and it should be modeled to indicate the pre-conditions, assumptions, post-conditions, and effects of such capabilities. There are special requirements for defining CPMS- interfaces for engineering and runtime invocation. Quality of services, including reliability (availability), real-time criticality, safety criticality, and security should be specified as well. Figure 4 shows a concept model of CPMS.

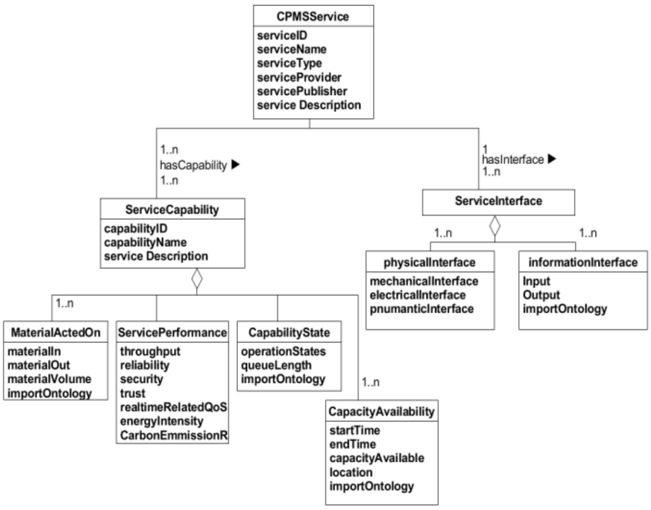


Fig. 4. CPMS Concept Model

Common Cyber-Physical Manufacturing Services

Among CPMS's non-functional properties, Service classification plays an important role for necessary service discovery. The use of taxonomy to describe the type of manufacturing services is inevitable. In general, common CPMS can be divided into three main categories: Processing, Transporting, and Storage. Processing services work on materials or partially

---

[13] http://dublincore.org/

finished products by changing their physical or chemical properties. Both transporting and storage services are logistical services to move material around or solely store it. Each of these services can be further classified as shown in Figure 5. For example, production services can be classified into batch production, continuous production, and discrete production. Discrete production includes Shaping services and Nonshaping services, etc. Additive manufacturing shapes material layer on layer. It can be classified further into seven categories based on ASTM F2792.

To provide an unambiguous identification of service classes and their relations, a standardized common service dictionary will be needed.

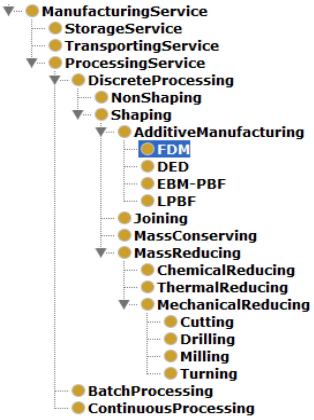


Fig. 5. An example of CPMS taxonomy

CPMS Capability Description

The taxonomy potentially defined by a common CPMS dictionary provides a hierarchy of concepts to describe service types. For accurate service match and composition, the capability of the service has to be explicitly modelled. The capabilities of a CPMS determine not only the range of material it can process, but also its ability to meet product manufacturing specifications, as well as capacity (throughput, failure rate) and availability requirements. In addition, the possible states of the capability and job queue information should be defined for run-time choreography and dynamic orchestration. For example, a capability may be identified as current, or may be identified for future times, as depicted in Figure 4. For every object property, an ontology can be associated for service matching provided by a meta-service.

CPMS Interface

Information about CPMS interfaces is necessary for fast re-engineering and dynamic CPMS orchestration or choreographic interactions. For re-engineering with physical adaptation using plug-n-play, physical interfaces are specified. The SmartFactoryFL project has defined unified mechanical and electrical interfaces for multi-vendor production module integration (Weyer et al. (2015)). For online automated service discovery-composition-execution, standard informational interfaces are necessary. Similar to web service description, an input message and an output message should be defined for service invocation. The exact binding methods should be specified as well.

4.2 *Manufacturing Activity/Goal Model*

Service requests are made when conducting manufacturing activities. Traditionally, manufacturing activity models define both information flows and physical processes involved in activities. When manufacturing activity models are used to represent service requests, the concrete process representation is not necessary. Instead, only the goals of the activity are modelled. A goal can be described as an objective to transform material or a product from one state to another state. The goal metamodel defined in WSMF is applicable in CPMS-based smart manufacturing systems. A goal class has non-functional properties similar to those of web services, e.g., type, name, etc. Its key object properties are the material states they are associated with, and other goals it has dependency on. Many existing works and standards potentially useful for manufacturing activity modeling exist, including Icam DEFinition for Function Modeling [14], Unified Modeling Language[15], Business Process Model and Notation[16]. Manufacturing activity models (Barkmeyer et al. (1998), Feng et al. (2015), Kang et al. (2001)) and common manufacturing goals potentially reduce the modeling effort. In addition, Cossentino et al. (2013) describes a process and tools to model goals in a Belief-Desire-Intention (BDI) framework which can be applied here.

4.3 *CPMS Reference Ontology*

A CPMS Reference Ontology captures the knowledge base to facilitate CPMS mediation – service query, discovery, goal deposition, goal and service match, service composition, etc. An ontology is a formal explicit specification of a shared conceptualization. Usually, an ontology defines two essential aspects – real-world concepts and formal semantics – necessary for information to be processed by a computer. A CPMS reference ontology defines basic concepts in the goal/activity and service domain and their relationships among the set of concepts. In addition, it provides a set of axioms for reasoning necessary for goal-service matching.

Many ontologies exist in the manufacturing domain that describe product, process, and resources (PPR) (Nielsen (2003)), and these can be reused here. The concepts of product and process link manufacturing goals with services, while the resource definition adds context to manufacturing service models by introducing the concepts of time and location. On the basis of the reference ontology, the mapping knowledge is necessary for describing each pair of two functional taxonomies from the service consumption domain and from the service provision domain. The mapping knowledge should consist of mappings between the concepts in the two functional taxonomies and their structural differences based on the taxonomies.

4.4 *CPMS Lifecycle Management*

The CPMS modeling framework can be applied to global production network resource planning, supply chain planning,

---

[14] http://www.idef.com/

[15] http://www.uml.org/

[16] http://www.bpmn.org/

shop-floor manufacturing re-engineering as well as on-the-fly production line reconfiguration. Because production facilities live much longer than the production period for a particular product, CPMS usually are designed to be adaptive. While CPMS type and interface tend to remain constant, service capability may change as equipment degrades and available capacity varies. Therefore, CPMS lifecycle management is very important for SOA-based SM systems. In addition to versioning information, service usage logs, and snapshots of performance history should be kept and reported by a service management system.

5. CASE STUDY – 3D PRINTER AS A MANUFACTURING SERVICE

To illustrate CPMS and its modeling framework, we describe a three-dimensional (3D) printing application scenario. We can imagine a "customized vehicle manufacturing" factory which provides 3D printing services for automotive components. This factory floor has all sorts of additive manufacturing machines, each of which can print parts from a range of feedstock materials and within a certain build volume. Big parts can be divided into several partitions and printed in parallel. Consequently, additional joining services may be required to produce final parts. By applying CPMS in the semantic framework, an order manager can invoke several 3D printing services, automated guided vehicle services, robot services, and welding services and aggregate them into new customer services in a semi-automatic way. In this scenario, we focus only on the description of a 3D printing service and illustrate it based on a concrete implementation.

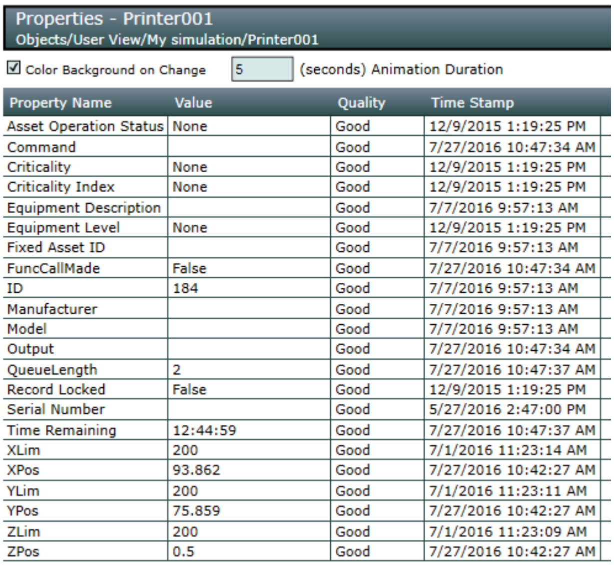
Our implementation is specifically devoted to exploring the capabilities an OPC UA-compliant data modeling and industrial automation software, Status Enterprise, in the context of implementing a distributed network of manufacturing machines to accomplish a common service-oriented goal. To evaluate and illustrate the CPMS modeling framework, we developed a mock-up use case – a 3D printer as a manufacturing service. Using such a service-oriented implementation, the 3D printing service can be accessed over the network with explicit capability representation, standard interfaces, grounding mechanisms, and non-functional features.

5.1 System setup

The architecture is based on B-Scada Status Enterprise, which is a visualization/automation software designed for industry. It provides an OPC UA-compliant interface and can characterize asset information and communication among assets using data models. As shown in Figure 6, the data model exposes all the non-functional features for the 3D printer, including the basic equipment description (model, ID, manufacturer) and service statuses (e.g., Time Remaining, QueueLength, tool positions). Specifically, a queue is constructed to handle concurrent requests from multiple clients/customers. If a job is sent to a busy 3D printer, it will be stacked into the queue and the QueueLength will be increased by one accordingly. TimeRemaining property keeps track of the current job's progress and all the remaining jobs in the queue to provide a real-time estimate of the total remaining time to a new job requester.

5.2 Process Execution

To implement these functionalities for the service invocation, a C# program is written as an SOA interface to the OPC UA server designed to mimic the functionality of a 3D printer and used as a platform to communicate with other SOA machines. Three major threads in the program are taking new requests, executing existing requests, and updating the remaining time. The detailed process execution is laid out as follows.



Properties - Printer001
Objects/User View/My simulation/Printer001

☑ Color Background on Change 5 (seconds) Animation Duration

Property Name	Value	Quality	Time Stamp
Asset Operation Status	None	Good	12/9/2015 1:19:25 PM
Command		Good	7/27/2016 10:47:34 AM
Criticality	None	Good	12/9/2015 1:19:25 PM
Criticality Index	None	Good	12/9/2015 1:19:25 PM
Equipment Description		Good	7/7/2016 9:57:13 AM
Equipment Level	None	Good	12/9/2015 1:19:25 PM
Fixed Asset ID		Good	7/7/2016 9:57:13 AM
FuncCallMade	False	Good	7/27/2016 10:47:34 AM
ID	184	Good	7/7/2016 9:57:13 AM
Manufacturer		Good	7/7/2016 9:57:13 AM
Model		Good	7/7/2016 9:57:13 AM
Output		Good	7/27/2016 10:47:34 AM
QueueLength	2	Good	7/27/2016 10:47:37 AM
Record Locked	False	Good	12/9/2015 1:19:25 PM
Serial Number		Good	5/27/2016 2:47:00 PM
Time Remaining	12:44:59	Good	7/27/2016 10:47:37 AM
XLim	200	Good	7/1/2016 11:23:14 AM
XPos	93.862	Good	7/27/2016 10:42:27 AM
YLim	200	Good	7/1/2016 11:23:11 AM
YPos	75.859	Good	7/27/2016 10:42:27 AM
ZLim	200	Good	7/1/2016 11:23:09 AM
ZPos	0.5	Good	7/27/2016 10:42:27 AM

Fig. 6. Data model of the OPC UA server for 3D printer

First, the connection between the server and the 3D printer is established. Through an OPC UA data connector, the specific values from the printer are connected to specific fields on the server when setting up the data connector on the server. Then a thread for taking requests is started. This thread polls the server for a service request, queues it for processing, and resets the input fields. It runs in the background for the entire duration of the program. As each request is queued, the time estimation is determined and added to a separate queue, as well as to the current time estimation. After that, a separate thread is created to write the time remaining in the queue to the server. This thread automatically updates the total remaining time for all the jobs in the queue, including the one being processed. Each command is dequeued and executed in sequence, updating the relevant server values (queue length, output field). After each job is completed and dequeued, the time estimate is recalculated based on a queue of the time estimations.

5.3 Discussion

In moving such a CPMS modeling framework into the real manufacturing world, two major issues arise: one is the challenge of handling multiple concurrent client requests, and the other one is automatic device registration for multiple machines in a scaled-up network. In this mock-up use case, the developed platform proved capable of handling concurrent requests from multiple clients. In addition, each client can access information from the server without interference from other server-client connections. These consequently warrant further investigation into the synthesis of an SOA system based on Status Enterprise.

As for the automatic device and service registration in a machine network, we adopt the concept of Devices Profile for Web Services (DPWS), which can enable secure web service messaging, discovery, description, and eventing on resource-constrained endpoints. Under the DPWS framework, a client can automatically detect other DPWS-enabled devices on a network, and discover and invoke the service functionality that each device or several devices can offer, according to their corresponding service types and service capabilities. Based on the CPMS taxonomy and ontology, manufacturing goals are decomposed into (or composed of) services that will be fulfilled by the DPWS devices automatically discovered in the network. Once an event is in progress, the devices are able to communicate through messaging and eventing functionalities to possibly reconfigure the system or adjust the process. Meanwhile, the service requester or customer can access the job information and get real-time feedback on the manufacturing progress.

6. CONCLUSIONS

With the introduction of smart devices embedded with more intelligence and predictive analytics, production units and logistic components are more accessible as services on a network. This paper described a service-oriented smart manufacturing architecture based on cyber-physical manufacturing services. Specifically, the paper proposed a semantic modeling framework that would enable easy development, usage, and dynamic composition of CPMS services. Under the framework, we identified capability gaps requiring further research effort in order to facilitate the adoption of smart manufacturing technology and the proposed architecture, including creation of standards for communication protocols, data model, knowledge representation, and CPMS characterization. Our future work will focus on cyber-manufacturing service models. Uses cases of applying CPMS to smart manufacturing systems will be studied and classified. Requirements on CPMS service models will be derived from the use cases. Existing manufacturing information modeling standards and ontologies will be used as bases to model CMPS capabilities, performance, interfaces and interaction methods.

DISCLAIMER

Any mention of commercial products is for information only; it does not imply recommendation or endorsement by NIST.

REFERENCES

Barkmeyer, E. J., (1998), SIMA Reference Architecture Part 1: Activity Models, NISTIR 5939.

Cossentino, M., Dalle Nogare, D., Giancarlo, R., Lodato, C., Lopes, S., Ribino, P., Sabatucci, L., and Seidita, V., (2014). GIMT: A tool for ontology and goal modeling in BDI Multi-Agent Design, *Workshop "Dagli Oggetti agli Agenti*. Deen, S.M. (Ed.), Agent-Based Manufacturing – Advances in the Holonic Approach, Springer-Verlag, Heidelberg, Germany, 2003.

Diep, D., Alexakos, C. and Wagner, T., (2007). An ontology-based interoperability framework for distributed manufacturing control, *IEEE Conference on Emerging Technologies and Factory Automation*, Patras, Greece.

Domingue, J.; Roman, D. and Stollberg, M., (2005). Web Service Modeling Ontology (WSMO): an ontology for Semantic Web Services. *W3C Workshop on Frameworks for Semantics in Web Services*, 9-10 June 2005, Innsbruck, Austria.

Feng S., Witherell, P. W., Ameta, G., Kim D. B., (2015), Fundamental Requirements for Data Representations in Laser-based Powder Bed Fusion, *Proceedings of the 2015 Manufacturing Science and Engineering Conference*.

Fensel, D. and Bussler, C., "The Web Service Modeling Framework WSMF", Electronic CommerceResearch and Applications, Vol. 1, Issue 2, summer 2002, Pages 113-137.

Hendler, J., (2011). Why the Semantic Web will never work, *8th Extended Semantic Web Conference (ESWC)*, Heraklion, Greek.

Kulvatunyou, B., Ivezic, N. and Srinivasan, V., "On Architecting and Composing Engineering Information Services to Enable Smart Manufacturing", J. Comput. Inf. Sci. Eng. 2016; 16(3).

Krafzig, D., Bank, K. and Slama, D., "Enterprise SOA: Service-Oriented Architecture Best Practices", Prentice Hall, 2004

Lemaignan, S., Siadat, A., Jean-Yves Dantan, J., and Semenenko, A., (2006). MASON: A Proposal For An Ontology Of Manufacturing Domain, *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, Prague, Czech Republic.

Lu, Y., Morris, K. C., Frechette, S. P., (2016a) " Current Standards Landscape for Smart Manufacturing Systems ", NISTIR 8107.

Lu, Y., Reddick, F., and Ivezic, N., (2016b). The Paradigm Shift in Smart Manufacturing System Architecture, *Advances in Production Management Systems*, Iguassu Falls, Brazil.

Marik, V., McFarlane, D., and Valckenaers, P., (Eds.), Holonic and Multi-Agent Systems for Manufacturing, Springer-Verlag, Heidelberg, Germany, 2003.

Nielsen, J., (2003). *Information Modeling of Manufacturing Processes: Information Requirements for Process Planning in a Concurrent Engineering Environment*, Doctoral Dissertation, Royal Institute of Technology.

Theorin, A., Ollinger, L., and Johnsson, L., (2012). Service-oriented Process Control with Grafchart and the Devices Profile for Web Services, *4th IFAC Symposium on Information Control Problems in Manufacturing,* Bucharest, Romania.

Theorin, A., Ollinger, L., and Johnsson, L., (2014). Service Orchestration with OPC UA in a Graphical Control Language, *9th IEEE International Conference on Emerging technologies and Factory Automation,* Barcelona, Spain

Vyatkin, V., Dai, W., Dubinin, and V., Christensen, J. H., (2014). IEC 61499 Implementation of Service Oriented Architecture: Case Study, *IEEE Conference on Robotics and Automation*, Hong Kong.

Weyer, S., Schmitt, M., Ohmer, M., and Gorecky, D.,(2015). Towards Industry 4.0 - Standardization As the Crucial Challenge for Highly Modular, Multi-Vendor Production Systems, *IFAC Symposium on Information Control in Manufacturing,* Ottawa, Canada.