Industry 4.0: An Overview of Key Benefits, Technologies, and Challenges

Chapter · May 2017

DOI: 10.1007/978-3-319-50660-9_1

2 authors:

Lane Thames
Tripwire, Inc.

Dirk Schaefer
University of Lincoln

Industry 4.0: An Overview of Key Benefits, Technologies, and Challenges

Lane Thames and Dirk Schaefer

Abstract A new revolution known as Industry 4.0 is occurring where countless elements comprising industrial systems are being interfaced with internet communication technologies to form the smart factories and manufacturing organizations of the future. Industry 4.0 and its associated technologies are currently being driven by disruptive innovation that promises to bring countless new value creation opportunities across all major market sectors. However, existing Internet technologies are plagued by cybersecurity and data privacy issues that will present major challenges and roadblocks for adopters of Industry 4.0 technologies. Industry 4.0 will face traditional cybersecurity issues along with its very own unique security and privacy challenges. If these challenges are not appropriately addressed, the true potential of Industry 4.0 may never be achieved. This chapter provides a brief overview of several key Industry 4.0 technologies and paradigms in order to give the reader a better understanding of the cybersecurity aspects of the remaining chapters in the book.

1 Introduction: Background and Motivation

A transformative event known as Industry 4.0 is occurring where countless elements comprising industrial systems are being interfaced with internet communication technologies to form the smart factories and manufacturing organizations of the future. Industry 4.0 and its associated technologies such as cloud-based design and manufacturing systems, the Internet of Things, the Industrial Internet of Things, and Social-Product Development are currently being driven by disruptive innovation that promises to bring countless new value creation opportunities across all major market sectors. However, existing Internet technologies are plagued by cybersecurity and data privacy issues that will present major challenges and roadblocks for

L. Thames (✉)
Tripwire, Inc., Atlanta, GA, USA
e-mail: lthames@tripwire.com

D. Schaefer
University of Bath, Bath, UK
e-mail: d.schaefer@bath.ac.uk

L. Thames and D. Schaefer (eds.), *Cybersecurity for Industry 4.0*,
Springer Series in Advanced Manufacturing, DOI 10.1007/978-3-319-50660-9_1

adopters of Industry 4.0 technologies. Industry 4.0 will face traditional cybersecurity issues along with its very own unique security and privacy challenges. If these challenges are not appropriately addressed, the true potential of Industry 4.0 may never be achieved.

A significant obstacle faced by those in Industry 4.0 related to cybersecurity is that of integration and cooperation amongst the stakeholders of any given Industry 4.0 organization. The core of this obstacle is that of language. Particularly, Industry 4.0 environments are made of diverse technologies spread across many disciplines with many different types of subject matter experts, but there are few standards and processes designed to assist each entity to speak a "common language" that appropriately aligns necessary objectives related to cybersecurity. For example, on the manufacturing side we have control engineers working with Operational Technology (OT). Similarly, on the Information Technology (IT) side we have system administrators working with traditional IT assets such as servers and software. On the one hand, a control engineer when dealing with securing OT assets is mostly concerned with 'mission assurance'. On the other hand, an IT system administrator is concerned with 'information assurance'. These objectives rarely align with one another. For example, a control engineer doesn't care about data loss over human life or machinery loss whereas a system administrator would never think about air gapping his battery backup units (UPS) for his servers. The drivers underlying these cybersecurity objects are vast and very different across domains. However, Industry 4.0 demands that these systems be integrated across all dimensions.

A primary goal of this book is to shed light on these aforementioned types of obstacles, needs, technologies and such as related to Industry 4.0 and cybersecurity. As alluded to in the previous paragraph, when approaching this subject, stakeholders need to understand the bigger picture. As such, the purpose of this chapter is to provide the reader with an overview of key technologies and paradigms related to Industry 4.0. The remainder of the book will emphasize cybersecurity aspects of Industry 4.0.

2 Industry 4.0 and Smart Manufacturing

Industry 4.0 is sometimes referred to as the 4th industrial revolution, and it is a vision of smart factories built with intelligent cyber-physical systems. It will enable manufacturing ecosystems driven by smart systems that have autonomic self-properties, for examples self-configuration, self-monitoring, and self-healing. Industry 4.0 will allow us to achieve unprecedented levels of operational efficiencies and accelerated growth in productivity. New types of advanced manufacturing and industrial processes revolving around machine-to-human collaboration and symbiotic product realization will emerge.

Industry 4.0 will encompass numerous technologies and associated paradigms. A few of these emerging paradigms include the Industrial Internet and the Industrial Internet of Things along with new 21st century product development paradigms such as cloud-based design, cloud-based manufacturing, crowd sourcing, and open innovation to name a few. A brief overview of these paradigms is provided in the following sections.

2.1 Industrial Internet and the Industrial Internet of Things

A new revolution is occurring within industry. It is a revolution resulting from the convergence of industrial systems with advanced computing, sensors and ubiquitous communication systems. It is a transformative event where countless industrial devices, both old and new, are beginning to use Internet Protocol (IP) communication technologies. We refer to this new revolution as the Industrial Internet of Things. The Industrial Internet of Things is a subset of what we have come to know as the Internet of Things (IoT). The IoT is an abstract idea that captures a movement that started when we began integrating computing and communication technology into many of the "things" that we use at home and work. It started with the idea of tagging and tracking "things" with low cost sensor technologies such as radio frequency identification (RFID) devices. However, the paradigm shifted as the market began delivering low-cost computing and Internet-based communication technologies, simultaneously with the rise of the ubiquitous smartphone.

This perfect storm of low cost computing and pervasive broadband networking has allowed the IoT to evolve. Now, the IoT includes all types of devices ranging from home appliances, light bulbs, automation systems, watches, to even our cars and trucks. Technically speaking, the IoT is a collection of physical artifacts that contain embedded systems of electrical, mechanical, computing and communication mechanisms that enable Internet-based communication and data exchange.

The Industrial IoT follows the same core definition of the IoT, but the things and goals of the Industrial IoT are usually different (see Fig. 1). Some examples of the things of the Industrial IoT include devices such as sensors, actuators, robots, manufacturing devices such as milling machines, 3D-printers, and assembly line components, chemical mixing tanks, engines, healthcare devices such as insulin and infusion pumps, and even planes, trains, and automobiles. Indeed, it is a vast spectrum of devices.

One interesting aspect of Industrial IoT devices is system complexity. In particular, Industrial IoT devices can contain systems of IoT systems. For example, an industrial robot as a whole might contain multiple sensors working both independently and as a group, and one or more of these sensors could control one or more actuators that, in turn, control the robots movement. Further, the sensors, actuators, and other parts of the robot can connect independently to an IP network with some centralized server that governs the overall control of the robot. Another term commonly used when discussing the Industrial IoT, and in particular, the Industrial Internet, is operation technology. Operation technology (OT) refers to the traditional hardware and software systems found within industrial environments. Some examples include programmable logic controllers (PLC), distributed control systems (DCS),

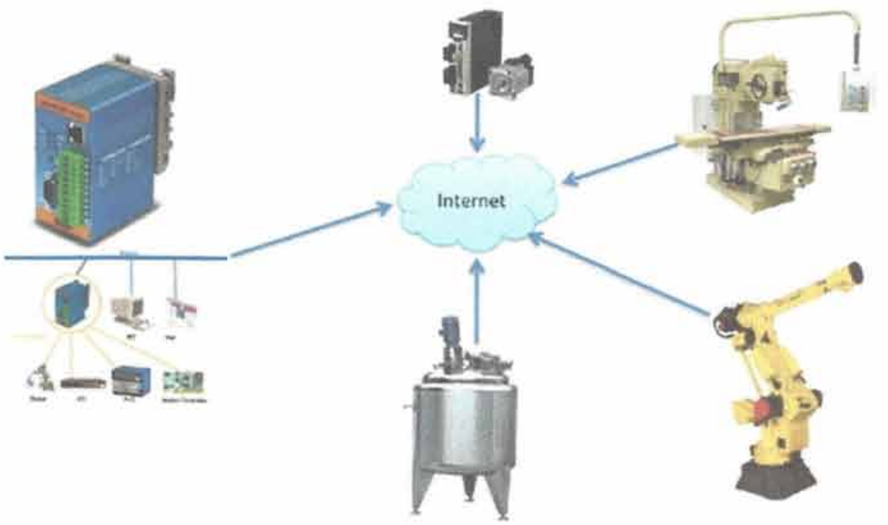

Fig. 1 Abstract idea of the industrial IoT

and human-machine interfaces (HMI). These systems are also known as Industrial Control Systems (ICS) because they control the various processes that occur within an industrial environment.

The Industrial IoT is a subset of the more general IoT. Hence, some of their characteristics are similar. The most common characteristic is that they all contain embedded computing and communication technology. These systems are largely focused on sensor technology along with the collection, transmission, and processing of sensory data. Communication is obviously a key component of the Industrial IoT. As illustrated by Fig. 2, the Industrial IoT can use both wired and wireless

Fig. 2 An example of industrial IoT communication architecture

communication. Some of the protocols used by Industrial IoT devices include Ethernet, Wi-Fi, WiMax, LR-WPAN, 2G/3G/4G telephony, IPv4, IPv6, 6LoWPAN, HTTP, CoAP, MQTT, XMPP, DDS, and AMQP, Profinet, ModBus, and DNP. There are different protocols for different use cases, commonly driven by environmental factors and resource constraints. For example, HTTP and MQTT are application layer protocols. HTTP, the hyper-text transport protocol, is a text-based protocol commonly used by web-based systems, i.e., web servers. It is a good protocol for client-server communications when there is more of a need to do only one-way data pulling. Although multiple sets of data packets moving in both directions are required for a client to pull down a web page from a server, the protocol is designed for pure client-server architectures. However, it is common for IoT devices to act as both client and servers. In these cases, HTTP is more difficult to implement, although it can be done using a polling methodology. MQTT was designed specifically for industrial network environments. It is a publish-subscribe messaging protocol, which eases the pain in terms of two-way communications where a device might act as both a client and server. Further, it is a light weight protocol in terms of transmission overhead, and it was designed to support lossy data transmission networks.

The Industrial Internet of Things will drastically change the future, not just for industrial systems, but also for the many people involved. If we can achieve the full potential of the Industrial IoT vision, many people will have an opportunity to better their careers and standards of living because the full potential of this vision will lead to countless value creation opportunities. This always happens when new revolutions get set into motion. The full potential of the Industrial IoT will lead to smart power grids, smart healthcare, smart logistics, smart diagnostics, and numerous other smart paradigms. For example, the Industrial IoT is at the heart of a related movement called Industry 4.0. Industry 4.0 is sometimes referred to as the 4th industrial revolution, and it is a vision of smart factories built with intelligent cyber-physical systems. It will enable manufacturing ecosystems driven by smart systems that have autonomic self-* properties such as self-configuration, self-monitoring, and self-healing. This is technology that will allow us to achieve unprecedented levels of operational efficiencies and accelerated growth in productivity. New types of advanced manufacturing and industrial processes revolving around machine-to-human collaboration and symbiotic product realization will emerge. It will truly be amazing to see all of the many benefits and technological advances that can be gained if we can achieve the full potential of this technology.

The Industrial Internet of Things can have a bright and shiny future. However, the devil is in the details. The number one challenge faced by the Industrial IoT is security and privacy. Cybersecurity and data privacy issues present major hurdles and roadblocks for adopters of Industrial IoT technologies. If we cannot alleviate many of the security and privacy issues that impact the Industrial IoT, we will not be able to achieve its full potential.

2.2 New 21st Century Product Development Paradigms

The force of globalization has served to instantaneously connect people from all across the globe, bringing with it game-changing opportunities to share knowledge and expertise to benefit in a collective manner (sometimes called share-to-gain). Friedman (2005) explains that the latest globalization phase, which he coins Globalization 3.0, began around the year 2000 and was enabled by the expansion of the internet on a global basis during the dot-com boom. According to Friedman, Globalization 3.0 is defined by individuals and small groups from across the globe collaborating in areas once dominated by less-connected western economies.

Tapscott and Williams (2008) explain that the advent of the internet has led to the development of cooperative collaboration networks, resulting in a power-shift from the once mighty hierarchical business model. These traditional business models, according to the authors, can no longer sustain successful innovation: "In an age where mass collaboration can reshape an industry overnight, the old hierarchical ways of organizing work and innovation do not afford the level of agility, creativity, and connectivity that companies require to remain competitive in todays environment." Simply put, industry is going to have to rethink the traditional models of business operation, as the amount of internal expertise they hold is dwarfed by that held by the global mass of peoples connected through globalization.

In academia and industry, the Pahl and Beitzs (1988) systematic design approach and Suhs (2001) Axiomatic Design theory are two of the most widely accepted design methodologies. Pahl and Beitz describe the product development process as a series of core transformations, from problem description to requirements list, to principal solutions and working structures, to preliminary design, to detailed layouts, and to final layout, form/dimensions, and manufacturing specifications. The design activities are classified into: product planning, conceptual design, embodiment design, and detail design. Suhs Axiomatic Design is a systematic design methodology based on matrix methods to analyze the transformation of customer needs into functional requirements, design parameters, and process variables.

However, neither Pahl and Beitzs design method nor Suhs Axiomatic Design theory offers a framework that facilitates seamless information, knowledge, and resource sharing, or aids participants of global value co-creation networks in identifying potential collaboration partners or resource providers (Franke et al. 2006). For example, value can be co-created when the participants of such networks identify information, knowledge, and manufacturing resources that are more cost effective than existing ones. The motivation of the research presented in this chapter is to bridge the gap between traditional product development methods and new methods that are required in the globalized world in which paradigms such as crowd-sourcing, mass collaboration and social product development are the order of the day. We begin by giving an overview of these paradigms.

In light of a continuing globalization alluded to above, product development is not only becoming increasingly complex and dynamic but also significantly more competitive. More and more of the skills and industries that traditionally fueled the economic prosperity of our nation are becoming the commodities of today and tomorrow. In addition, new product development paradigms and associated competencies required to successfully compete in the "flat" world are emerging at a mind-boggling rate of speed. Some of these new paradigms can be considered real game changers and are worth a closer look.

Complex social networks, consisting of millions of individuals, have formed over the Internet through emerging Web 2.0 technologies such as blogs, discussion boards, wikis, and collaboration networks such as Facebook or LinkedIn, video networks such as YouTube, and countless others. Information on almost anything is readily available to everyone through the Web, anytime and anywhere. Individuals, who have never met physically, are already collaborating on the development of complex products and services for major companies, collectively solving challenging problems that are openly "crowd sourced" to a community of interested engineers, scientists, and even hobbyists. While this may sound weird to some of us, for the next generation of engineers, it will be the norm. Their number one material to work with will be information, their final product(s) will be intellectual property and innovation, and their generation is already becoming known as the generation of knowledge workers.

Globalization 3 has led to the emergence of various game-changing paradigms anticipated to foster breakthrough innovation. These paradigms are characterized by the self-organization of individuals into loose networks of peers to produce goods and services in a very tangible and ongoing way. These paradigms include, among others, crowd-sourcing, mass collaboration, and open innovation. Enabling technologies for these paradigms include first and foremost the Internet, social networking platforms for business, cloud computing, as well as new business philosophies, such as "share to gain". New organizational structures based on self-organizing communities are emerging to complement traditional hierarchies. According to Tapscott and Williams (2008), new principles for success in the globalized world are (a) openness to external ideas, (b) individuals as peers, (c) sharing of intellectual property, and (d) global action. In such emerging organizations, individual success is defined by the recognition gained through contributions towards a common goal rather than by following the directions from the top management. An organization's success is determined by its ability to integrate talents of dispersed individuals and other organizations.

Crowd sourcing is defined as "the act of sourcing tasks traditionally performed by specific individuals to a group of people or community (crowd) through an open call" (Wikipedia 2017). Because it is an open call to a group of people, it attracts those who are most fit to perform tasks or solve problems, and provide fresh and innovative ideas. This way, a significantly higher talent pool than the one any company could possibly have can be tapped. Procter & Gamble, for example, created their own platform for this, called Connect + Develop, years ago.

Closely related to crowd sourcing is the paradigm of mass collaboration. Here, the idea is to harness the intelligence and ideas of many (or the crowd), to find innovative solutions to complex problems. Mass collaboration can be defined as "a form of collective action that occurs when large numbers of people work independently on a single project, often modular in its nature. Such projects typically take place on the Internet using social software and computer-supported collaboration tools such as wiki technologies, which provide a potentially infinite hyper-textual substrate within which the collaboration may be situated" (Wikipedia 2017). While the online encyclopedia Wikipedia may be one of the most prominent examples for a mass-collaborative project, there are many other examples of projects related to the development of real world products in this fashion.

The two preceding paradigms are considered to foster Open Innovation, a term coined by Henry Chesbrough (2003). According to his definition, open innovation is "a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as the firms look to advance their technology". He also states that "...the central idea behind open innovation is that in a world of widely distributed knowledge, companies cannot afford to rely entirely on their own research, but should instead buy or license processes or inventions (i.e. patents) from other companies. In addition, internal inventions not being used in a firms business should be taken outside the company (e.g. through licensing, joint ventures or spin-offs)". This is closely related to what others refer to as share-to-gain. Crowd sourcing, mass collaboration and open innovation certainly have a number of appealing characteristics. However, there are two major issues that currently make companies shy away from these new paradigms. One is intellectual property (IP), which can be tricky waters to navigate, especially on a global level. The second one is a lack of new business models to go along with the new paradigms. Companies still need to make money, and while everyone will agree that putting together an online encyclopedia in a share-to-gain fashion is a neat thing to do, designing and manufacturing, for example, cars and airplanes that way isnt quite that straight forward.

The technical and enabling backbones for these new paradigms are the Web and the Internet, which has grown into a huge "supercomputer" that is continuously getting smarter, i.e., capable of responding to its semantic surroundings, for the world to share. Today, a myriad of software packages to facilitate all sorts of online collaboration, both for professional as well as personal purposes, are available. They range from simple video communication tools such as Skype to more complex collaboration suits like Wiggio, up to full-blown product design solutions, such as Dassault Systems CATIA V6 in concert with their cloud-based collaboration platform SwYm.

Cloud computing, originally conceptualized in the 1960s, is a fancy marketing term for networked computers that provide services (or resources) through the Internet to a network of clients who utilize them, usually on a pay-as-you-go cost model. The three most prominent cloud computing application areas are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Clouds can be public, private, or a hybrid in nature. In other words, companies may choose to implement their own internal cloud as a Local Area Network (private cloud), use the cloud-infrastructure from a third-party provider (public cloud), or opt for a hybrid for example, to rent and run software as a service in the public cloud and store application data in a local, private cloud.

Recently, cloud computing has made its advent to the domain of computer-aided product development. In addition to running CAD systems as a service in the cloud, other business-related everything-as-a-service models have started to emerge. One such model relates to manufacturing and aims at utilizing physical resources, for example, 3D printers for additive manufacturing, mills, lathes, and other manufacturing-related equipment, through the cloud. Long-term, computer-aided product development in general (including design, analysis and simulation, as well as manufacturing) is anticipated to become predominantly cloud-based. It is a promising new model to facilitate globally distributed design and manufacture processes that seamlessly integrate both virtual and physical resources. In the next section, we provide a discussion of cloud-based design and manufacturing (CBCM) that seeks to enhance the Industry 4.0 paradigm by harnessing the power of crowd-sourcing, open innovation, and mass collaboration along with technologies such as cloud computing, the Internet, and the web as a new 21st Century Product Development Paradigm.

3 Cloud-Based Design and Manufacturing

Before introducing CBDM and identifying its key characteristics, we first review some of the existing definitions of cloud computing:

- Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (NIST 2011).
- Cloud computing refers to both the applications delivered as services over the internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what we will call a Cloud (Armbrust et al. 2010).
- Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms, and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the infrastructure provider by means of customized SLAs (Vaquero et al. 2009).
- A cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers (Buyya et al. 2008).
- Cloud computing is both a UX and a business model. It is an emerging style of computing in which applications, data and IT resources are provided to users as services delivered over the network. It enables self-service, economies of scale

> and flexible sourcing options an infrastructure management methodology—a way of managing large numbers of highly virtualized resources, which can reside in multiple locations...(IBM 2010).

From above, a number of well-known and widely cited definitions of cloud computing are presented. Here, we put these ideas in a historical perspective in order to understand the origin of cloud computing, where it comes from, and its evolution. While the term cloud computing was only coined in 2007, the concept behind cloud computing, delivering computing resources through a global network, was rooted in 1960s (Licklider 2010). The term "Cloud" is often used as a metaphor for the Internet, and refers to both hardware and software that deliver applications as services over the Internet (Armbrust et al. 2010). When looking backward, one realizes that cloud computing is based on a set of pre-existing and well researched concepts such as utility computing, grid computing, virtualization, service oriented architecture, and software-as-a-service (Bohm 2010). One milestone is utility computing, proposed by John McCarthy in 1966. The idea of utility computing is that "computation may someday be organized as a public utility". Due to a wide range of computing related services and networked organizations, utility computing facilitates integration of IT infrastructure and services within and across virtual companies (Parkhill 1966). Another milestone is that Ian Foster and Carl Kesselman proposed the concept of grid computing in 1999. A computational grid refers to a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities (Foster and Kesselman 1999). Since cloud and grid computing share a similar vision, Foster et al. (2008) identified the main differences between grid computing and cloud computing. The greatest difference is that cloud computing addresses Internet-scale computing problems, utilizing a large pool of computing and storing resources, whereby grid computing is aimed at large-scale computing problems by harnessing a network of resource-sharing commodity computers, dedicating resources to a single computing problem.

Compared to grid computing, we envision that cloud computing would be the most promising underlying concept that can be borrowed in the fields of design and manufacturing due to the advantages of greater flexibility, ubiquitous availability of high capacity networks, low cost computers and storage devices as well as service-oriented architecture. Thus, before exploring CBDM in more detail, it is worthwhile to take a close look at what make cloud computing unique and how it is being leveraged in design and manufacturing fields.

Cloud computing can be seen as an innovation from different perspectives. From a technical perspective, it is an advancement of computing history that evolved from calculating machines with binary digit systems, to mainframe computers with floating-point arithmetic, to personal computers with graphical user interfaces and mobility, to the Internet that offers computing resources via distributed and decentralized client-server architectures, and eventually to utility, grid, and cloud computing (Boem et al. 2010). From a business perspective, it is a breakthrough which is changing the mode of IT deployment and potentially creating new business models.

In order to leverage cloud computing in existing manufacturing business models and enterprise information systems, cloud manufacturing, based on cloud computing and service-oriented technologies, is proposed (Tao et al. 2011). The architecture, core enabling technologies, typical characteristics for cloud manufacturing, and the difference and relationship between cloud computing and cloud manufacturing has been discussed. Xu (2012) discusses the potential of cloud computing that can transform the traditional manufacturing business models by creating intelligent factory networks. Two types of cloud computing adoptions in the manufacturing sector have been suggested, direct adoption of cloud computing technology in the IT area and cloud manufacturing where distributed resources are encapsulated into cloud services and managed in a centralized manner.

4 Defining Cloud-Based Design and Manufacturing (CBDM)

Based on the concept of cloud computing, we propose a definition of CBDM as follows (Wu et al. 2012):

Cloud-Based Design and Manufacturing refers to a product realization model that enables collective open innovation and rapid product development with minimum costs through a social networking and negotiation platform between service providers and consumers. It is a type of parallel and distributed system consisting of a collection of inter-connected physical and virtualized service pools of design and manufacturing resources (e.g., parts, assemblies, CAD/CAM tools) as well as intelligent search capabilities for design and manufacturing solutions.

Figure 3 illustrates the concepts underlying the foundations and principles of CBDM systems aligned with our proposed definition thereof. At this point, it is noteworthy to explain the use of the term Cloud. Communication and network engineers have traditionally encapsulated the inherent interconnection complexity of networks with cloud diagrams. In essence, a network of any reasonable size is too complex to draw on a diagram. Consequently, cloud diagrams are used to hide the interconnect complexity while simultaneously revealing the primary details of a particular network diagram. As seen from Fig. 3, the Internet communication cloud forms the basic and required underlay network for any CBDM system in general. As stated previously, CBDM technologies are enabled by Internet-based information and communication technologies. This dependency is represented by illustrating CBDM as an overlay in Fig. 3. Moreover, Fig. 3 seeks to illustrate the overall and basic interconnectivity of the primary elements of a CBDM system. For example, the human resources of a CBDM system form their own human-centric network, which is represented by design teams, social networks, and students, just to name a few. Likewise, the cloud resources, which include human, virtual, and physical resources, are

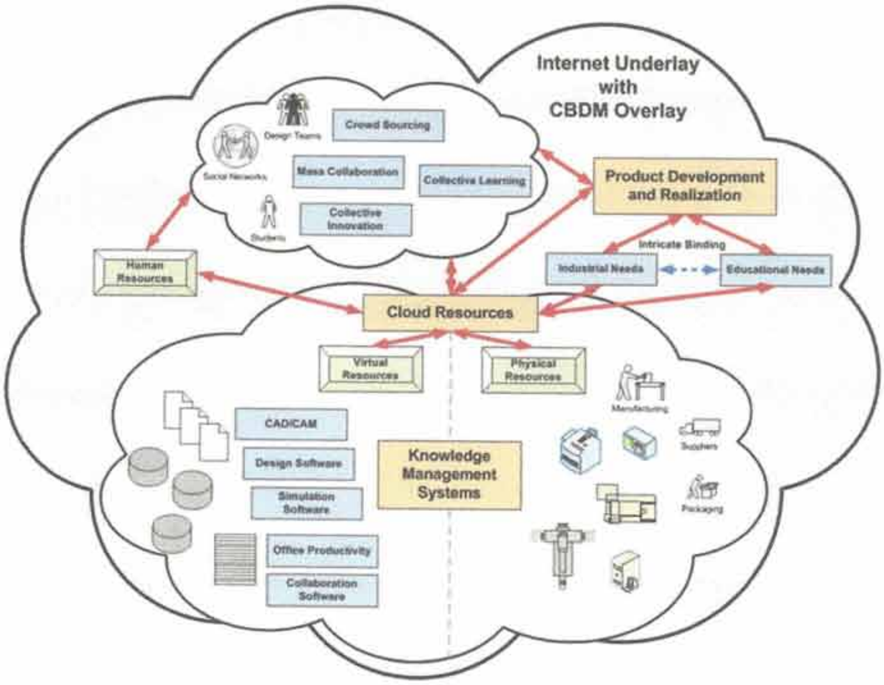

Fig. 3 The CBDM concept

illustrated along with their appropriate partitions. One of the primary goals of CBDM is to enable efficient product development and realization processes. Hence, appropriate interconnections are established between this goal and the basic partitions of the diagram. Further, one should observe the needs of the product development and realization process: namely, industrial needs and educational needs. These two sectors comprise the basic categories of entities that need the CBDM functionality. Moreover, industrial needs and educational needs are, in general, intricately bound. Industry will use CBDM technology to produce raw goods and services. Obviously, industry depends on educational entities for the following: (1) to educate students on the basic principles and foundations of CBDM systems in order to accomplish their economic goals and (2) to conduct cutting-edge research and development on the underlying details of CBDM systems. Hence, the educational and industrial entities are intricately bound.

In addition, the essential characteristics of CBDM, including on-demand self-service, ubiquitous network access, rapid scalability, resource pooling, and virtualization are emphasized as prerequisites to enable CBDM as follows:

- On-demand self-service: A customer or any other individual participating in the cloud can provide and release engineering resources, such as design software, manufacturing hardware, as needed on demand. It provides a platform and intuitive, user-friendly interfaces that allow users (e.g., designers) to interact with other users (e.g., manufacturers) on the self-service basis.

- Ubiquitous network access: There is an increasing need for a so-called customer co-creation paradigm, which enables designers to proactively interact with customers, as well as customers to share different thoughts and insights with designers. In order to easily reach such a communication media, it requires a capability of broad and global network access. The CBDM systems can provide such access to the network where cloud consumers reside through multiple tools, e.g., mobile phones and personal digital assistants. CBDM allows various stakeholders (e.g., customers, designers, managers) to participate actively throughout the entire product realization process.
- Rapid scalability: The CBDM systems allow enterprises to quickly scale up and down, where manufacturing cells, general purpose machine tools, machine components (e.g., standardized parts and assembly), material handling units, as well as personnel (e.g., designers, managers, and manufacturers) can be added, removed, and modified as needed to respond quickly to changing requirements. It helps to better handle transient demand and dynamic capacity planning under emergency situations incurred by unpredictable customer needs and reliability issues. For example, the cloud system allows the cloud service consumers to quickly search for and fully utilize resources, such as idle and/or redundant machines and hard tools, in another organization to scale up their manufacturing capacity.
- Resource pooling: The cloud providers design and manufacturing resources are pooled to serve cloud consumers in a pay-per-use fashion. Resources include engineering hardware (e.g., fixtures, molds, and material handling equipment) and software (e.g., computer-aided design and Finite Element Analysis (FEA) program packages). The CBDM model enables convenient and on demand network access to such a shared pool of configurable manufacturing resources. The real time sensor inputs, capturing the status and availability of manufacturing resources, ensures effective and efficient cloud resource allocation.
- Virtualization: The CBDM systems provide a virtual environment through the simulation of the software and/or hardware upon which other software runs. It enables enterprises to separate engineering software packages, computing and data storage resources from physical hardware, as well as to support time and resource sharing.

4.1 Cloud Based Design

Cloud Based Design (CBD) is a part of the CBDM concept with a focus on design aspects. CBD refers to a design model that leverages Web 2.0 (i.e., social network sites, wikis, online reviews, and recommender systems) and Web 3.0 to support the gathering, representation, processing, and use of product design-related information that is distributed across social media and the Internet (Wu et al. 2013).

Traditionally, it has been assumed that generating design ideas and implementing them was the exclusive task of design teams. However, CBD has the potential to enable customers, engineers, and other participants to share information through social media by integrating Web 2.0 tools into product design processes. For example, a Web 2.0 site provides service providers and consumers a vehicle to communicate and interact with each other through online product reviews. In this way, designers can easily get feedback on their customers user experience.

In addition, due to the vast amount of product design-related data in social media, engineers are facing a significant challenge in quickly find the information they need. Web 3.0 allows the information to be precisely described in terms of ontology that can be understood by machines. Web 3.0 will support effective and efficient discovery, automation, and reuse of data for CBD.

4.2 Cloud Based Manufacturing

Cloud based manufacturing (CBM) is the other part of the CBDM concept with a focus on the manufacturing aspect. CBM refers to "a customer-centric manufacturing model that exploits on-demand access to a shared collection of diversified and distributed manufacturing resources to form temporary, reconfigurable production lines which enhance efficiency, reduce product lifecycle costs, and allow for optimal resource loading in response to variable-demand customer generated tasking". (Wu et al. 2013) the motivation for introducing CBM is based on the belief that CBM can lead to important advances in new ways of conducting manufacturing activities from the following perspectives.

First, one of the main reasons for the adoption of CBM by manufacturing enterprises is the emerging outsourcing and crowd sourcing models in manufacturing. CBM may (1) facilitate Small and Medium-Sized Enterprises (SMEs) run manufacturing operations more cost effectively by utilizing excessive manufacturing resources owned by large enterprises; and (2) enable large sized enterprises to develop and enhance their core competencies and innovation capabilities by crowd-sourcing labor-intensive tasks.

Second, one of the distinguishing characteristics of CBM is that CBM allows enterprises to quickly scale up and down, where manufacturing cells, general purpose machine tools, machine components (e.g., standardized parts and assembly), material handling units, as well as personnel (e.g., designers, managers, and manufacturers) can be added, removed, and modified as needed to respond quickly to changing requirements.

4.3 CBDM Services

Figure 4 presents some example CBDM cloud services available to a cloud consumer.

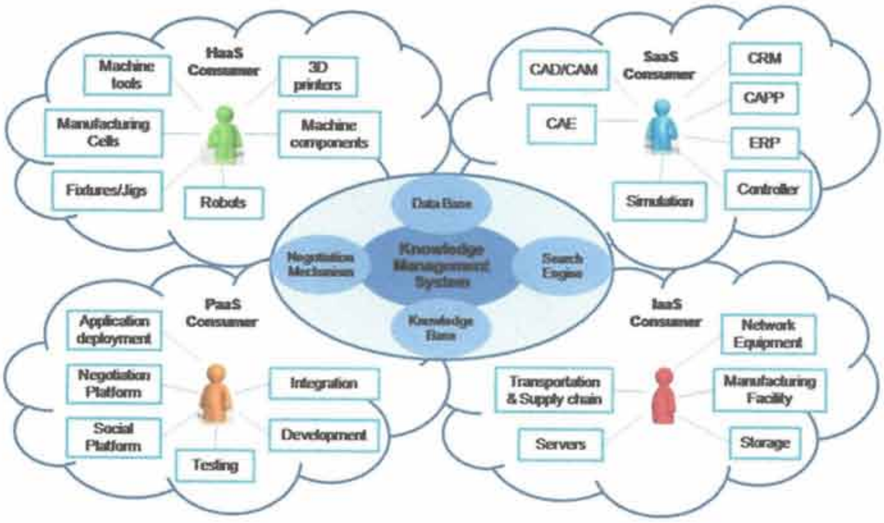

Fig. 4 CBDM example services

Hardware-as-a-Service (HaaS): HaaS delivers hardware sharing services, e.g., machine tools, hard tooling, and manufacturing processes, to cloud consumers through the CBDM system. Cloud consumers are able to rent and release hardware provided by a third party without purchasing them. The Cubify.com 3D online printing service is a good example, which allows cloud consumers to produce parts through any mobile device using their online 3D printing service without purchasing 3D printers. The consumers of HaaS could be either engineers or end users, who may utilize manufacturing hardware.

Software-as-a-Service (SaaS): SaaS delivers software applications, e.g., CAD, CAM, FEA tools, and Enterprise Resource Planning (ERP) software to cloud consumers. Cloud consumers are able to install and run engineering and enterprise software through a thin client interface without purchasing full software licenses. The cloud service offered by Dassault Systems and Autodesk are by far the best known examples among engineering analysis applications, allowing remotely running 3D software and high performance discrete computing environments (Autodesk 2017; Dassault 2017). The consumers of SaaS can be designers, engineers and managers, who need access to software applications.

Platform-as-a-Service (PaaS): PaaS provides an environment and a set of tools (e.g., an interactive virtual social platform, a negotiation platform, and a search engine for design and manufacturing solutions) to consumers and application developers to assist them in integrating and delivering the required functionality. A good example is Fujitsu, providing a high-speed thin client environment, server consolidation, and license consolidation, which dramatically reduces manufacturing costs and development times by leveraging a knowledge base in the cloud (Fujitsu 2017).

Infrastructure-as-a-Service (IaaS): IaaS provides consumers with fundamental computing resources, e.g., high performance servers and storage space. These services are offered on a pay-as-you-go basis, eliminating downtime for IT maintenance as well as reducing costs dramatically. The consumers of IaaS could be engineers and managers, who need access to these computing resources.

5 CBDM: A First Generation Implementation

Generally speaking, the integration of Information Technologies systems (IT) with Operation Technology (OT) systems is crucial for the success of Industry 4.0. This is true from a core technology perspective, and IT/OT integration also happens to be a key challenge for the cybersecurity aspects of Industry 4.0. We will discuss the cybersecurity aspects of IT/OT integration for Industry 4.0 later in this chapter. For now, we will discuss it within the context of core Industry 4.0 technologies and how it applies to our real world experience while implementing our first generation CBDM system.

Figure 5 illustrates the ideas of IT and OT convergence. What does it mean to converge IT with OT? First, let's consider the underlying components. Operation technology refers to the traditional hardware and software systems found within

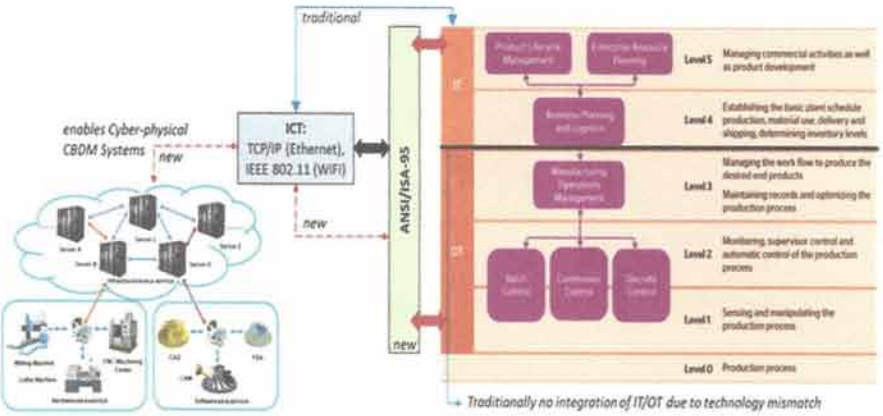

Fig. 5 The idea of IT and OT convergence

industrial environments. Some examples include programmable logic controllers (PLC), distributed control systems (DCS), and human-machine interfaces (HMI). These systems are also known as Industrial Control Systems (ICS) because they control the various processes that occur within an industrial environment. Information technology generally refers to the software, servers, personal computers, mobile phones, and such that comprise the business side of an organization. Convergence of IT and OT relates to the interconnection of these systems using modern networking technologies such as Ethernet and IP networking. You see, traditionally, OT systems where 'air gapped' from IT networks. OT systems commonly used proprietary and/or specialized communication protocols and didn't come equipped with networking that was compatible with those of the IT networks. Recently, however, OT systems have rapidly been adopting standards such as WiFi, Ethernet, and IP networking. As such, many enterprises have been connecting OT systems with their IT systems. There are many different schools of thought as to whether this is good or bad. Regardless, having OT systems connected to our IT systems is rapidly becoming the norm and will drive future cyber physical designs for Industry 4.0 stakeholders.

The development of our first generation CBDM prototype was achieved while working on a DARPA research project called Manufacturing Experimentation and Outreach (MENTOR). MENTOR was a sub-project of the DARPA Adaptive Vehicle Make (AVM) program. The overall objectives of these research programs were to aid designers in managing system complexity and in reducing product development time and effort. Our research and development efforts for MENTOR was to develop a prototype infrastructure that served as an enabling platform for the deployment of a variety of programmable manufacturing equipment, such as 3D printers, to over 1000 high schools throughout the country and to orchestrate a series of prize-based challenges to encourage competition and collaboration within high school teams as they design and build cyber-electro-mechanical systems. Figure 5 once again illustrates the basics of our IT/OT integration challenge. The left side of the figure reveals our overall objectives: the need to integrate manufacturing equipment along with software such as CAD programs into an IT architecture. The right side of the figure reveals the functional levels across the IT and OT sides. The figure does not reveal, however, the complexity of the challenge we faced with MENTOR. You see, IT and OT integration within a **single** organization is very challenging. The integration of IT and OT systems across many diverse organizations (i.e., 1000+ high schools) is exponentially challenging due to differing IT and OT governance issues across the independent organizations. As such, we had to develop an systems model and CBDM system that would enable seamless integration of IT and OT systems across organizational boundaries. We solved this issue using a hybrid distributed-centralized system model that formed the basis of our first generation CBDM system. The next few sections describe our model, architecture, and implementation details.

5.1 An Infrastructure for Distributed Collaborative Design and Manufacturing Inspired by the Cloud Computing Paradigm

In general, an infrastructure is a system of assets such as physical components, human resources, operational processes, and organizational structures required to facilitate a particular set of outcomes. For example, a countrys transportation infrastructure facilitates the delivery of raw goods, in which raw goods are used to produce products, in which products are then delivered to consumers. Naively, one might assume that the transportation infrastructure consists simply of a countrys network of roadways. However, the transportation infrastructure is more complex than just the roadway network. Instead, it consists of the roadway network system, the system of organizations producing raw goods, the system of organizations who produce products from the raw goods, the organizations who deliver the products and raw goods, and the consumers of the final product. It is easy to argue that an infrastructure is a complex System of systems. One particular concept common to any infrastructure is that the infrastructures system of assets are employed for the purpose of combining problem holders with problem solvers to produce some set of outcomes that facilitate the solution for the underlying need implied by the necessity of the infrastructure. An infrastructure is a collection (system) of assets that collectively produce a set of desired outcomes, which would not be attainable by any particular asset alone. The value added by the infrastructure is determined by the interconnection of its assets, which is the interconnection between problem holders and problem solvers.

We have developed a distributed infrastructure with centralized interfacing system (DICIS) model for CBDM, which is illustrated in Fig. 6. The components within DICIS include all user interfacing components (i.e. web browsers), communications and security components (the Internet and enterprise firewall systems), human assets (users, producers, consumers, managers, etc.), and the actual Manufacturing process assets. Note that manufacturing process assets (MPA) include software components such as CAD tools and packages as well as physical components such as 3D printers, milling machines, electrical prototyping boards, and robotic equipment. Even though a "pure" cloud computing framework normally only represents software systems, the DICIS model for our CBDM includes both virtual resources (i.e. software, computer hardware, etc.) as well as physical and human resources such as the equipment listed above. In essence, the DICIS model and its implementation as a CBDM system can be viewed as an integrated design and manufacturing infrastructure, which can support industrial applications as well as educational needs such as computer-centric laboratory coursework and research.

The DICIS model categorizes CBDM assets into three primary groups: (1) Human Assets, (2) Communication Assets, and (3) Manufacturing Process Assets. Further, human, communication, and manufacturing process assets are bound to both the centralized interface (CI) and the distributed infrastructure (DI). The distributed infrastructure incorporates the primary physical, virtual, and human resources of the

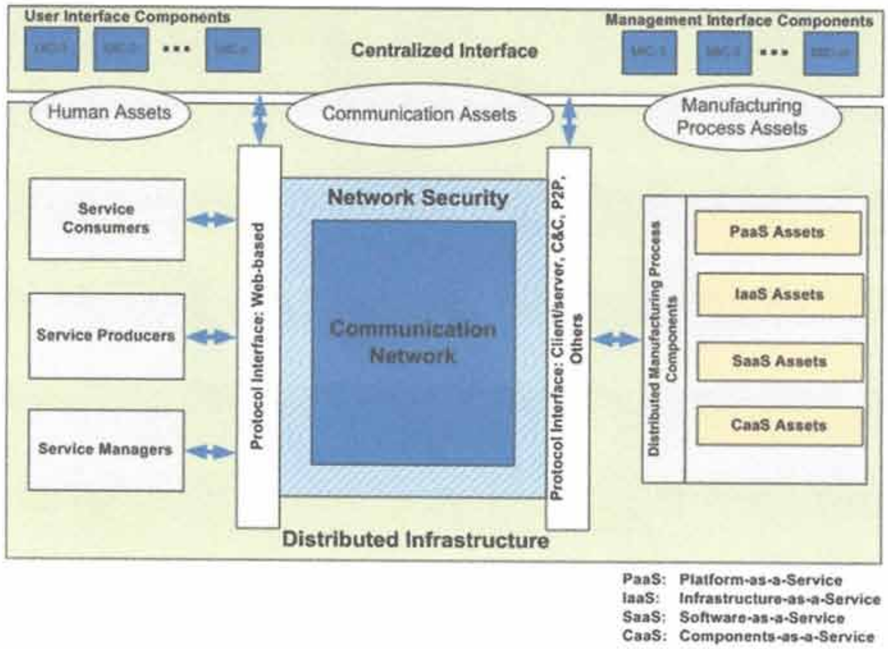

Fig. 6 The DICIS model for CBDM

CBDM. However, the centralized interface, which includes two primary groups of components referred to as the user interface components (UIC) and management interface components (MIC), provides the resources that glue the system together.

The DICIS model considers three human asset categories: (1) service consumers, (2) service producers, and (3) service managers. Service consumers utilize the services offered by the CBDM. Service consumers include, for example, students participating in distributed design and manufacturing projects, researchers/engineers investigating a new design prototypes, or companies with geographically distributed manufacturing shops that need to manufacture the components of a new product. Service producers provide human resources in term of intellectual capital and labor that result in provisioning of useful services. For example, a laboratory assistant or production manager could be a service producer who installs a new set of devices and equipment into the CBDM and integrates these components to form a new consumer service. An example could be a remote manufacturing site that is installing a new 3D printer and milling machine into the CBDM that should be used by human assets (consumers) of the CBDM. Service managers administer the various resources in the CBDM, depending on the scope of their management roles. Service managers perform operations such as creating new user accounts, assigning user roles, scheduling projects, installing new CBDM resources, and scheduling system maintenance, just to name a few.

In the most general sense, service producers and service managers are problem solvers, whereas service consumers are problem holders. However, service producers and service managers can be problem holders that seek services of other service producers and service managers. Further, a particular user can simultaneously be a service consumer, producer, and/or manager, depending on the users role with respect to the system as a whole. For example, consider the user Alice. Alice can be a student participating in project A, a producer for project B, and a manager of project C.

The communication assets of DICIS are comprised of four primary components: (1) communication network, (2) network security, (3) human asset service communication interface (SCI), and (4) manufacturing process asset service communication interface. We assume that the communication network is based on the Internet Protocol (IP) such that standardized, ubiquitous, Internet-based communications take place. The network security component encapsulates the communication network component, which reflects the idea that securability is needed but also that in modern day enterprise networks, it already exists in several forms, but most notably in the form of firewall systems. In order to capitalize on the ubiquitous web, the human asset SCI uses web based protocols. Using web based protocols such as the HyperText Transport Protocol (HTTP) between human assets and the centralized interface will minimize CBDM deployment costs as it removes the need to develop specialized interface software for system utilization. However, the manufacturing process asset SCI can be more diverse, and different protocols such as client-server, command and control, and peer-to-peer protocols can be used, depending on the particular requirements of a given subset of the CBDM.

The manufacturing process assets of the DICIS model consist of hardware (physical) and software (virtual) design and manufacturing resources. Our current CBDM under investigation, which is an implementation of the DICIS model, consists of a heterogeneous hardware and software environment, and it supports manufacturing and laboratory hardware devices such as milling machines, lathes, laser cutters, 3D printers (3DP), and do-it-yourself (DIY) 3D printers.

For the software systems, our CBDM utilizes various computer-aided manufacturing (CAM) technologies, which are software systems that convert digital models of parts designed by our integrated CAD tools into machine-based fabrication instructions. Moreover, we are developing a range of software applications for design and manufacturing activities, as well as system and resource management. Some of these software applications include the commercial Dassault Systems suite of design and analysis tools such as CATIA and Simulia, which enable high-end CAD and analysis capabilities, as well as collaboration. We are also integrating various additive manufacturing tools into the system, such as tools for locating and utilizing 3D printers within the DICIS network.

5.2 A CBDM Workflow Example

A few basic details of our CBDM architecture are illustrated in Fig. 7. As shown in the figure, the CBDM system consists of a centralized interfacing server (CIS). The current version our CBDM uses a CIS platform that is based on the Sakai learning management system. From Fig. 7, several geographically dispersed users who are collaborating on a design project and are utilizing services of the CBDM such as CAD design tools, 3D printers, and CNC machines. The CIS also provides applications for resource management and scheduling. Once designs are ready for prototyping, STL files generated by the CAD tool are submitted to the CBDM 3D printing service framework. Further, for parts that are to be fabricated in metal, a design file (i.e., STL files) can be sent to a milling machine, which is controlled via software running on a milling machine PC (server), for the actual production of the end product. Note that the user interface is composed of web-browser interfaces into the CAD software as well as the 3D printing and milling machine controller software.

Figures 8 and 9 will be used to further explain the CBDM process. Figure 8 illustrates how our CBDM provides distributed and collaborative design and manufacturing services to three engineers. From Fig. 9, two of the engineers are working locally while the third is located at a distant site. Real-time collaboration is enabled via video telecollaboration services. Further, the three engineers are able to access the CAD design software, but not simultaneously. Instead, CAD control is transferred on-demand to any give designer in the collaborative design session by way of issuing a transfer input control request to the software application. Figure 9 shows how the design file from Fig. 8 is transferred to a remote 3D printer within the CBDM. In essence, once the collaborating engineers from Fig. 8 have completed their design and are ready to develop an AM prototype of the design, other software within the CBDM such as AM-Select is used to transfer design files from the CAD service to the 3D printer service.

6 Software Defined Cloud Manufacturing

Industry 4.0 and smart manufacturing of the future will, indeed, take advantage of numerous Internet-based technologies and associated paradigms. In the previous sections, we have described several key paradigms driven by Industry 4.0, including technologies such as cloud-based design and manufacturing systems that we have personally investigated over the past few years. An underlying objective of our research for the past few years is illustrated by Fig. 10. Particularly, one of our main research objectives, reflected by our research described above as well as what follows in this next few sections, is to enable globally distributed and collaborative cyber-physical product creation using cloud-based design and manufacturing, which in turn establishes an inter-relationship between the industrial Internet of Things

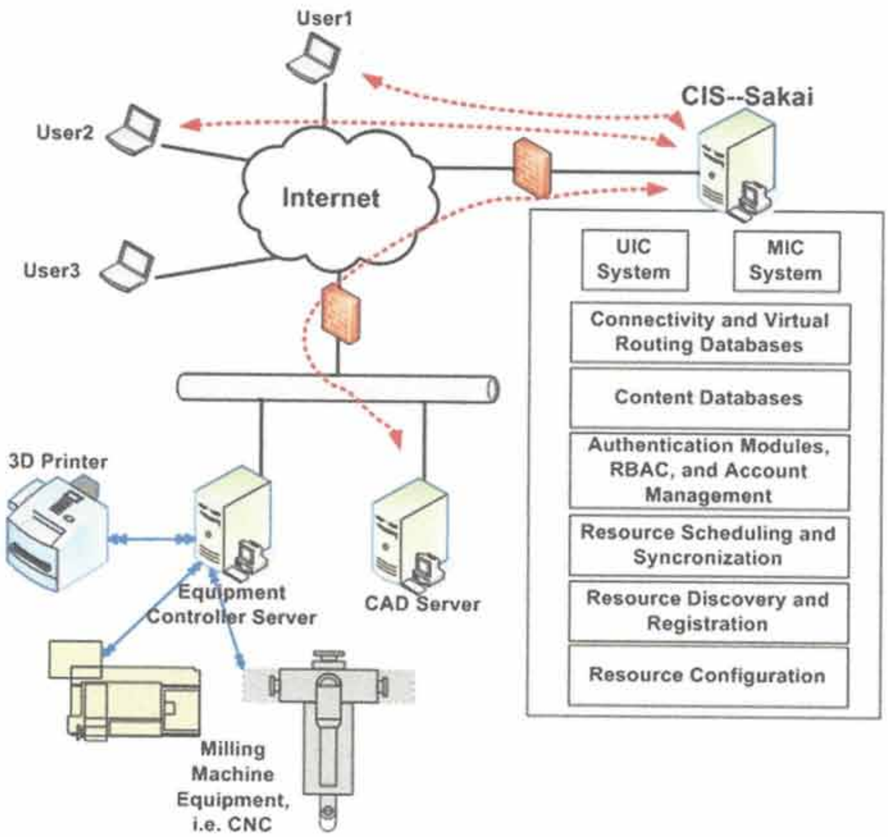

Fig. 7 CBDM workflow example

and the Internet of Services. We have tried to capture the ideas underlying these objectives in Fig. 10. The right hand side illustrates a CBDM system that is enabling cloud-based manufacturing and design services. This are essentially viewed from a consumer perspective as an Internet of Services enabled by an Industrial Internet of Things. As the figure reflects, design services can be globally distributed as well as the manufacturing services. It is the underlying CBDM framework that allows this abstraction. The 'consumer view' is that of using resources provided by an Internet of Services, which is composed of the actual entities (hardware entities, software entities, etc.) within the Industrial Internet of Things.

The objectives are aligned with the overall vision of Industry 4.0. However, the overall implementation is not trivial. Many challenges and open problems remain along the research path we have taken. This book seeks to address a fundamental problem faced by Industry 4.0, which is that of cybersecurity and is addressed by the remaining chapters in this book. However, many other challenges remain. In particular, system complexity and the ability to manage Industry 4.0 system complexity

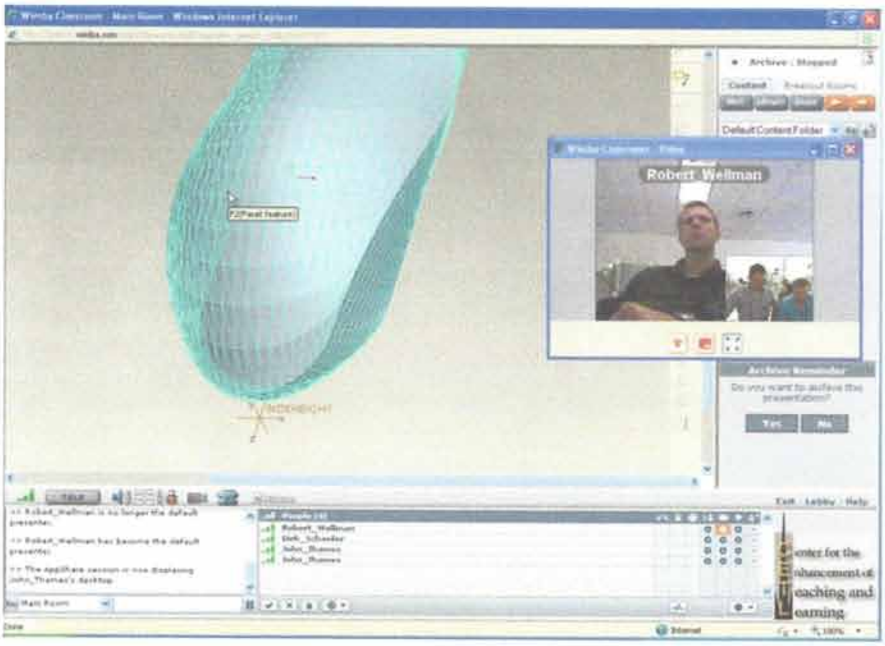

Fig. 8 Collaborative design via CBDM

is very important. In light of this, we have taken our research to a new level seeking to address complexity challenges, while also enabling cybersecurity functionality. Our new direction was guided by our first generation CBDM implementation as described above. Our latest advancements that seek to address these issues is based on an idea we refer to as Software Defined Cloud Manufacturing (SDCM). In the following sections, we will describe the ideas underlying SDCM.

6.1 Software-Defined Systems

Recently, the information technology field has begun to utilize software defined systems. It is a new paradigm of thinking about hardware and software, largely enabled by inexpensive, highly-functional hardware and virtualization technologies. Technologies that have emerged within this domain include software-defined networking, software-defined storage, software-defined computing, software-defined data center, and software-defined radio.

Software-defined networking (SDN) is defined as the physical separation of the network control plane from the forwarding plane where a control plane controls several devices. To understand the ramifications of this design, one must consider the paradigm it is replacing. Particularly, non-SDN networking devices are based on a

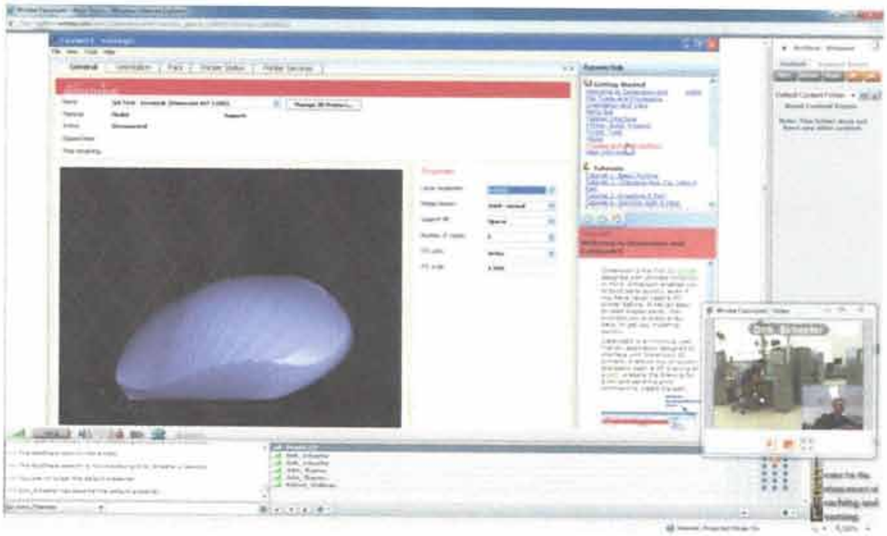

Fig. 9 Sending a design file to a CBDM 3D printer resource

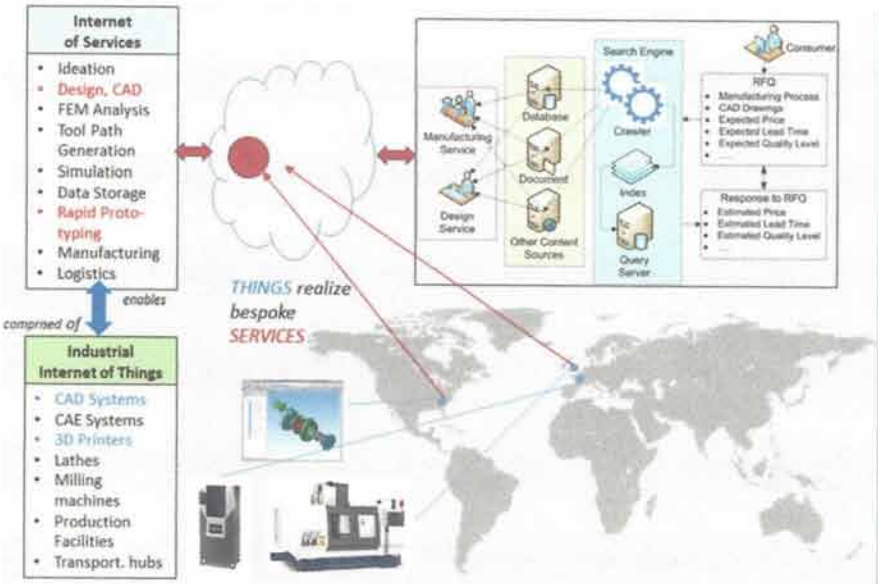

Fig. 10 Relationship between CBDM, the industrial internet of things, and the internet of services

design whereby each network device is totally isolated from the other devices in its network. Although it might coordinate and work with other devices, its so-called control plane is isolated to itself and its control plane functionality cannot be modified (outside of traditional patching, upgrades, etc.). With SDN, the control plane is managed centrally, it is defined by software, and it can apply to multiple devices. The idea is that network devices have generic hardware that does not require vendor-specific software, and the control plane functionality can be molded to fit a given design goal and can apply to multiple devices. SDN is known to be flexible, manageable, adaptive, and very cost-effective. It allows the control plane to be directly programmable instead of fixed software that is only configurable.

The software-defined supply chain is yet another emerging software-defined system recently described by Paul Brody (2013). Brody suggests that product design and manufacturing are changing, and the change is due to emerging, maturing, and converging technologies. Namely, Brody suggests that three particular technologies will reshape manufacturing. These technologies include 3D printing, next generation intelligent assembly robots, and open source hardware. Brody goes on to say that "Success in the future will require developing and adopting a new set of mental models, business processes, and enterprise technologies" (Paul Brody 2013).

6.2 A Software Defined Cloud Manufacturing Architecture

Recall from the previous sections that our first generation CBDM model and architecture was based on a hybrid distributed-centralized system. The architecture was driven by ideas included in the DICIS model. The architecture was composed of highly distributed design and manufacturing components. However, the entire system was 'glued' together using a centralized interfacing server. The reason for such a design was to remove IT/OT integration challenges across disparate organizations with different IT governance policies. To do this, a command and control communication (C2) architecture was developed whereby each entity in the collaboration network, i.e., all of the MENTOR high schools' hardware units such as 3D printers, where interfaced with a lightweight C2 software agent that established a TCP session with the CIS. The CIS was then in charge of interconnecting the various devices, software, and users of the collaboration network, i.e., the CBDM network. Basically, a single CIS server served as the core intelligence of the CBDM network. While this design works well for small networks, it does not scale. Moreover, the centralized nature of this design does not align with our overall vision of enabling CBDM as a fully distributed, global network of cloud-based resources.

Considering these issues, we went back to the drawing board and developed a new architecture that converted the CIS into a fully distributed cloud-based network of intelligent components (which you will see in a few minutes we call SDCM controllers). Our new architecture is modeled by software defined systems. In general, software-defined systems are characterized by properties such as being agile, programmable, manageable, configurable, interoperable, adaptable, and protectable. Indeed, Industry 4.0 technologies and smart manufacturing systems can benefit from these characteristics.

In what follows, we describe our new Software-Defined Cloud Manufacturing (SDCM) architecture to achieve these characteristics for various Industry 4.0 systems

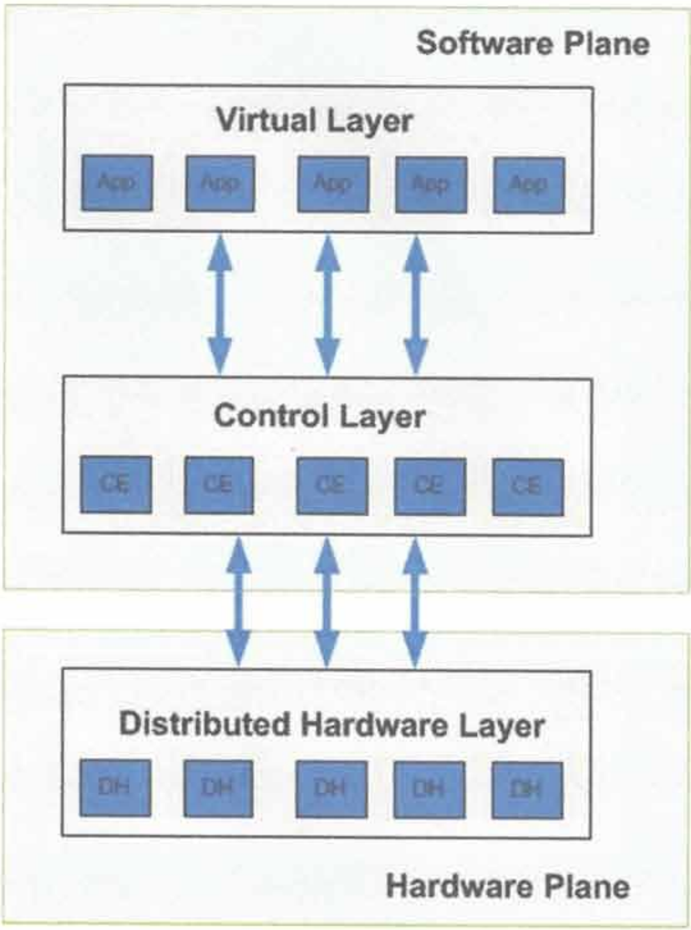

Fig. 11 A simplified software-defined cloud manufacturing architecture

in order to enable cyber physical product creation via a more distributed approach to CBDM. Our simplified SDCM architecture is illustrated in Fig. 11.

In this architecture, we assume a large network of hardware and software elements that have Internet-based communication frameworks, i.e. a TCP/IP stack. The goal is to utilize elements that constitute an Industry 4.0 system such as an IIoT, CBM, or SPD or combination thereof.

An important aspect of the SDCM architecture is separation of concerns (SoC). SoC is a design principle that allows one to break extremely large and complex systems into manageable parts. For example, the world-wide Internet is based on the SoC design principle. Our proposed SDCM architecture is first broken into two planes: the Software Plane and the Hardware Plane. In the architectures current state, we seek to distinguish the hardware elements from software elements. In particular, hardware does the final work whereas software will define how the work is orchestrated through to completion. The hardware plane includes a Distributed Hardware Layer (DHL). The DHL is further comprised of Distributed Hardware (DH) elements. For example, a DH could be a generic 3-D printer built on generic hardware from some particular maker community.

The software plane contains two layers, the virtual and control layers. The control layer is comprised of control elements (CE) and the virtual layer contains final user applications. Information flows are indicated by the arrows. The DHL communicates with the control layer and vice versa using an appropriate communication interface. Likewise, the virtual layer interfaces with the control layer.

Within each layer, multiple elements can be composed to create higher-level elements. As such, we define a software defined cloud manufacturing entity as a three-tuple $M = (V, C, D)$, where M is an SDCM entity, $V = \{a\}$, $C = \{ce\}$, $D = \{dh\}$. We say that V is a set representing an application composition, C is a set representing a control element composition, and D is a set representing a hardware composition. Particularly, our software defined cloud manufacturing model represents per-level element composition services that provide, in general, the capability to produce complex manufacturing services.

6.3 SDCM Domain Specific Configuration Language

Being 'software defined' immediately implies that things are defined by software. As such, we have developed a very light weight domain specific language that, at this time, serves more so as a configuration language than a pure programming language. The syntax was inspired by the VHDL hardware description language and the ideas underlying structured query language (SQL). The structure is given as follows:

BEGIN Doman_Specific_Language:
SELECT service **AS** "X"
TYPE: {"key":"value"}
END
END

The goal of this light weight language was to provide a simple machine readable language that enables an entity within the SDCM network to find some service that is available based on various types of characteristics. This basic language serves well in terms of a prototype implementation. However, during the next phase of our research, we will be incorporating a more established and more mature language. In particular, in our future work we will be adopting the Software Component Ensemble Language (SCEL) (Nicola et al. 2014). Observe that an SDCM composition can be viewed as an ensemble of components. It turns out the SCEL "is a language for programming service computing systems in terms of service components aggregated according to their knowledge and behavioral policies" (Nicola et al. 2014). We believe that SCEL has a perfect mapping to SDCM that will allow us to enhance our overall goals and objectives.

6.4 SDCM Workflow Scenarios

In this section, we will provide a short overview of how the SDCM architecture will work in practice. A key functionality of SDCM is composition. At "runtime" applications, control elements, and distributed hardware elements are dynamically composed, and the composition depends on the overall SDCM service being provided.

Control elements are responsible for composition. Composition is initiated via an applications invocation (residing at the virtual layer). Controller elements are the core masterminds of a given SDCM service and contain the controller logic for composing the elements of the given service.

Composition of hardware elements can be achieved across a vast spectrum of scenarios. Here, we provide an overview of the ideas of this composition process at two opposite ends of the spectrum. On one end of this spectrum, we consider the idea of design and manufacturing hardware that might be found in domains such as open source hardware, DIY hardware, and maker spaces. As Brody ?? suggests, 3D printing, next generation intelligent assembly robots, and open source hardware will have significant impact on future manufacturing processes. Open source hardware (and its associated open source software) will lead to fast and incremental updates to hardware platforms. This could be utilized by various manufacturing entities. One aspect of this scenario is the ability to reconfigure these hardware platforms based on a desired set of functionality; this is where SDCM comes in to the picture. Figure 12 will be used to illustrate the ideas.

From Fig. 12, there is an open source hardware platform (OSHP) that contains 3 high-level components (C1, C2, and C3). This OSHP is considered to be a single hardware element (DHE). It has manufacturing capabilities that can be initiated via the composition of component C1 with component C2 or with component C2 and component C3. In this example, a controller element is in charge of uploading and

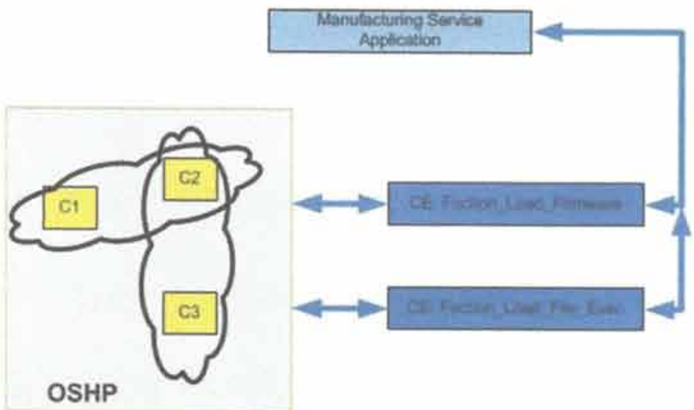

Fig. 12 An illustrated view of the SDCM workflow

installing firmware that configures the device based on the desired composition. After the firmware has been installed and initialized, a second control element loads a design file into the system and begins its particular manufacturing service.

The previous example exploited the ideas of open source hardware of the near future where resources included within a given hardware platform can be composed to produce a certain type of manufacturing service. Further, it assumes that the platform could be reconfigured for different types of manufacturing services. Obviously, this will depend on the hardware platform and its internal resources. The workflow that is described, however, can be utilized at a different level of abstraction. This workflow provided by SDCM can be applied to multiple, independent hardware platforms. This abstraction is indeed a powerful aspect of SDCM. An equivalent workflow, for example, could be the composition of a manufacturing robot (i.e., C1 is the robot) along with a 3D printer and CNC milling machine (C2 and C3, respectively). An SDCM manufacturing entity could be the composition of the robot and the CNC machine or the robot and the 3D printer. Controller elements and user interfaces (virtual applications) can be developed to implement a manufacturing service that utilizes these compositions to create some artifact. The artifact could in turn be designed by some cloud-based CAD program that is also brought in as a higher-order composition. For example, the CAD program can be yet another component in the total composition.

In the following scenarios, we will connect back to our first generation CBDM prototype in order for the reader to understand the evolution of our work in this area. First, let's review Fig. 13. During our work with MENTOR and our first generation CBDM prototype, we developed C2 communication interfaces that interconnected

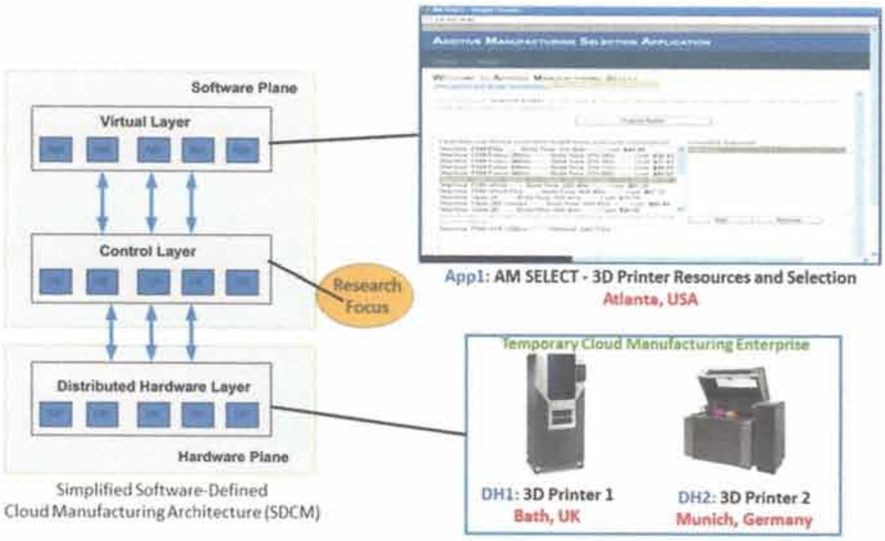

Fig. 13 SDCM and CBDM—high level

via the CIS all of the hardware and software entities in the network. Once such software application was called 'AM SELECT' (Additive Manufacturing Select) and it was used to allow users of the CBMD network to find additive manufacturing devices with the CBDM network. Once selected, based on types such as materials, cost, time to complete the job, etc., the CIS interconnected the user to an available printer. Figure 13 reveals how this is accomplished, at a high level, with our SDCM architecture. It is very similar. The AM SELECT application is located within the virtual layer and the 3D printers are located within the distributed hardware layer. However, the intelligence required to connect the pieces together (as well as all other functionality such as resource discovery, virtual routing, authentication, authorization, access control, etc.) is implemented at the control layer, which is a vast collection of control elements highly distributed throughout a cloud environment.

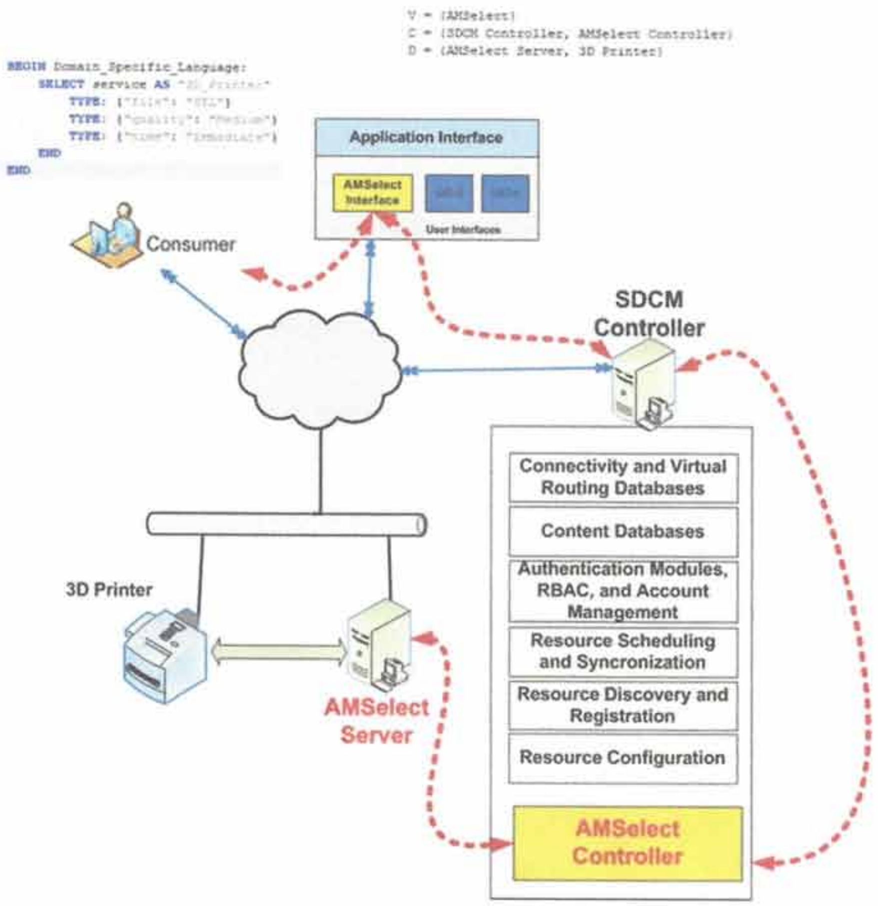

Fig. 14 SDCM and CBDM—low level

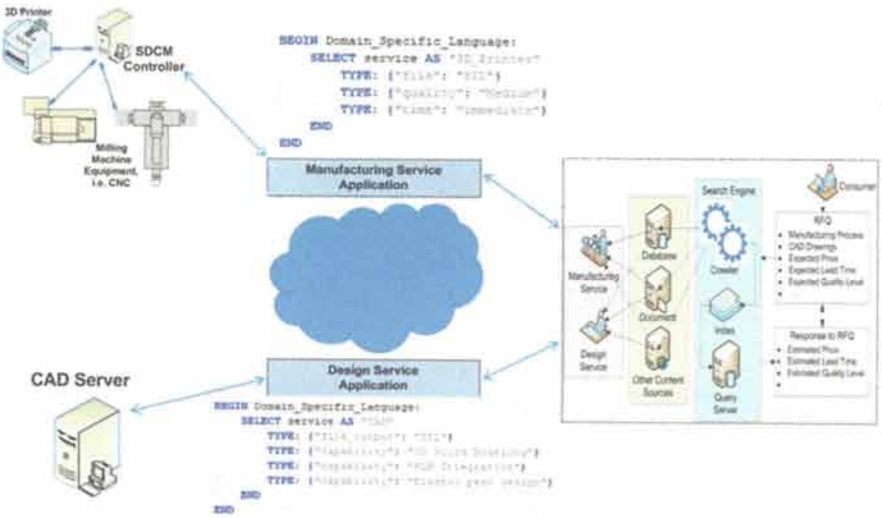

Fig. 15 SDCM and CBDM: enabling manufacturing and design services

Let us explain these concept further with Fig. 14. This figure is similar to Fig. 7. The CIS is replaced by a distributed collection of SDCM controllers. These controllers implement functions such as connectivity and virtual routing, content, resource configuration, and such. In Fig. 14 a SDCM consumer interacts with the AM SELECT interface. Upon selection of the appropriate additive manufacturing resource, the interface generates the domain specific language requesting a 3D Printer resource that accepts STL files, produces artifacts with medium quality, and is available immediately. This request gets sent to the controller cloud and gets processed. Once an appropriate 3D printer is located, an SDCM controller interconnects the printer to the AM SELECT interface for the user to submit the STL file.

We provide one last scenario, which is illustrated by Fig. 15. In this figure, we are simply trying to illustrate how we have tied all the pieces together. Our SDCM architecture is the underlying backbone that allows us to implement a fully distributed CBDM network. The CBDM network allows consumers and producers to interact with one another via design and manufacturing services.

7 Closure

In this chapter, we have tried to set the stage for the remainder of this book by introducing key technologies and paradigms that are driven by the overall vision of Industry 4.0. A large portion of the chapter was devoted to discussing Cloud-based Design and Manufacturing (CBMD) because of two reasons. First, the authors have been researching and implementing CBDM technology for over 5 years. Second, we have also studied the cybersecurity aspects of CBDM and felt that our in depth discussion of the topic would serve the reader well in understanding the remaining material in the book. In terms of CBDM, we have presented the big picture of how it has emerged as a new paradigm to support globally distributed design and manufacturing in the broader context of social product development and the so-called new industrial revolution (Anderson 2012). We have explained the underlying technical fundamentals of CBDM, our current extension to CBDM using a software defined cloud manufacturing architecture, and have presented a summary of our various implementations. Once again, the goal was to provide the reader with a broad background of Industry 4.0 technologies and paradigms. Now, however, we must turn our attention to the primary purpose of this book: cybersecurity. The remaining chapters of this book will address many aspects of cybersecurity and its need for advancing Industry 4.0 technologies.

References

Anderson C (2012) Makers: the new industrial revolution. Crown Business

Armbrust M, Fox A, Griffith R, Joseph A, Katz R, Konwinski A (2010) Above the clouds: a view of cloud computing. Commun ACM

Autodesk (2017). www.autodesk.com

Bohm M (2010) Cloud computing and computing evolution. Technical report

Brody P (2013) Get ready for software-defined supply chain. http://www.supplychainquarterly.com/topics/Manufacturing/20140110-get-ready-for-the-software-defined-supply-chain/

Buyya R, Yeo C, Venugopal S (2008) Market-oriented cloud computing: vision, hype, and reality for delivering IT services as computing utilities. CoRR

Chesbrough W (2003) Open innovation: the new imperative for creating and profiting from technology. Harvard Business Press

Dassault (2017). www.3ds.com

De Nicola R, Loreti M, Pugliese R, Tiezzi F (2014) A formal approach to autonomic systems programming: the scel language. ACM Trans Auton Adapt Syst

Foster I, Kesselman C (1999) The grid: blueprint for a new computing infrastructure. Morgan Kaufmann

Foster I, Zhao Y, Raicu I, Lu S (2008) Cloud computing and grid computing 360-degree compared. In: Grid computing environments workshop

Franke N, Von Hippel E, Schreier M (2006) Finding commercially attractive user innovations: a test of lead user theory. J Prod Innov Manage 23(4):301–315

Friedman T (2005) It's a flat world after all. New York Times 3:33–37

Fujitsu (2017). www.fujitsu.com

IBM (2010) Dispelling the vapor around cloud computing. ibm.com

Licklider J (2010) Topics for discussion at the forthcoming meeting, memorandum for: members and affiliates of the intergalactic computer network

NIST (2011) NIST cloud computing reference architecture—special publication 500–292

Pahl G, Beitz W (1988) Engineering design: a systematic approach. Springer

Parkhill (1966) The challenge of the computer utility. Addison-Wesley publication

Suh N (2001) Axiomatic design: advances and applications. Oxford University Press

Tao F, Zhang L, Venkatesh V, Luo Y, Cheng Y (2011) Cloud manufacturing: a computing and service oriented manufacturing model. J Eng Manuf 225(10):1969–1976

Tapscott D, Williams A (2008) Wikinomics: how mass collaboration changes everything. Portfolio trade

Vaquero L, Merino L, Caceres J, Lindner M (2009) A break in the clouds: towards a cloud definition. ACM Comput Commun Rev

Wikipedia (2017). www.wikipedia.org

Wu D, Greer M, Rosen D, Schaefer D (2013) Cloud manufacturing: drivers, current status, and future trends. In: ASME international manufacturing science and engineering conference

Wu D, Schaefer D, Rosen D (2013) Cloud-based design and manufacturing systems: a social network analysis approach. In: International conference on engineering design

Wu D, Thames L, Rosen D, Schaefer D (2012) Towards a cloud-based design and manufacturing paradigm: looking backward, looking forward. In: ASME international design engineering technical conference and computers and information in engineering conference

Xu X (2012) From cloud computing to cloud manufacturing. Robot Comput Integr Manuf 28:75–86