Published in final edited form as:
J Comput Inf Sci Eng. 2016 September ; 16(3): . doi:10.1115/1.4033725.

On architecting and composing engineering information services to enable smart manufacturing

Boonserm (Serm) Kulvatunyou,
Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, MD 20899-8260, USA

Nenad Ivezic, and
Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, MD 20899-8260, USA

Vijay Srinivasan
Fellow ASME, Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, MD 20899-8260, USA

Boonserm (Serm) Kulvatunyou: boonserm.kulvatunyou@nist.gov; Nenad Ivezic: nenad.ivezic@nist.gov; Vijay Srinivasan: vijay.srinivasan@nist.gov

Abstract

Engineering information systems play an important role in the current era of digitization of manufacturing, which is a key component to enable smart manufacturing. Traditionally, these engineering information systems spanned the lifecycle of a product by providing interoperability of software subsystems through a combination of open and proprietary exchange of data. But research and development efforts are underway to replace this paradigm with engineering information **services** that can be composed dynamically to meet changing needs in the operation of smart manufacturing systems. This paper describes the opportunities and challenges in architecting such engineering information services and composing them to enable smarter manufacturing.

1 INTRODUCTION

In a keynote paper at the 2nd International Through-life Engineering Services Conference, McMahon and Ball [1] addressed the role of information systems in improving the through-life support of long-lived, complex artifacts. They pointed out the promise offered by information systems in a number of areas including productivity and more accurate and responsive assessment of artifact conditions. They also stressed the need for understanding the complexity and interlinked nature of the engineering information involved in through-life engineering services.

Correspondence to: Boonserm (Serm) Kulvatunyou, `boonserm.kulvatunyou@nist.gov`.

In this paper, we delve a little deeper into the engineering information that is shared among different phases in a product's lifecycle and across its supply chain. We also explore how different aspects of the information can be offered as services. In particular, we examine the role of open engineering information and messaging standards that are relevant to through-life engineering services.

The idea of sharing engineering information as services is not new [2, 3]. When Web Services – supported by service-oriented architecture (SOA) – became a reality more than a decade ago, such engineering information services offered an attractive, alternative avenue to support integrating various engineering activities. Since then, two major developments have accelerated this trend. The first is the wide-spread digitization of the entire manufacturing sector that positions data and information[2] at the front and center of all modern manufacturing. This, in turn, has heightened the need for engineering information standards in the manufacturing sector. The second is the virtualization of computing and communication resources using 'clouds.' This has moved the engineering service functions to the clouds with several attendant opportunities and challenges.

We start by setting the stage with the modern digitization of manufacturing that will enable smart manufacturing, in Section 2. Section 3 positions several standards, some of which have been extensively updated recently, as exemplar enablers of smart manufacturing systems. Sections 4 to 6 discuss standards related to service-oriented architecture and cloud computing, which are positioned as an enabling infrastructure for smart manufacturing systems. In particular, Section 4 provides a brief introduction to service-oriented architecture. Section 5 proposes a cloud architecture model to help guide manufacturing enterprises in their cloud adoptions. Section 6 discusses opportunities and challenges based on a proposed, high-level cloud architecture, which is a synthesis of a number of industry-specific architectures. It is a blueprint to enable compositions of various engineering information and computing services for a diverse set of manufacturing-services scenarios, such as the one described in Section 7. Some concluding remarks are made in Section 8 after a brief summary.

2 Digitization of manufacturing

In April 2012, the Economist magazine published an influential article that proclaimed that the Third Industrial Revolution, in the form of the digitization of manufacturing, is well underway [4]. By its reckoning, the first industrial revolution began in Britain in the late 18th century, with the mechanization of the textile industry. The second industrial revolution came in the early 20th century, when Henry Ford mastered the moving assembly line and ushered in the age of mass production. In the third industrial revolution currently under way, manufacturing is going digital.

---

[2] The difference between data and information is often blurry. Data is typically collected directly from equipment or operation, and is regarded as raw and needing post-processing such as cleaning, transformation, and/or addition of meta-data/context. Once post-processed, it then becomes information (e.g., temperature is data, but temperature associated with a particular item and process from which it is collected becomes information). Software services are typically constructed to deliver data (i.e., data services) as well as and information (i.e., information services, which can be a composition of data and other analytical and transformation services). Information services are then composed to create smart services that deliver knowledge. Composition of information services is where the value is created.

A year later after the Economist article, using a slightly different counting method, the German manufacturing industry came up with the nickname Industrie 4.0 to refer to the current era in manufacturing [5]. By its count, the first three industrial revolutions came about as a result of mechanization, electricity, and information technology; now, the introduction of Internet of Things and Services into the manufacturing environment is ushering in a fourth industrial revolution called Industrie 4.0. The German manufacturing industry predicts that in the future, businesses will establish global networks that incorporate their machinery, warehousing systems, and production facilities in the shape of Cyber-Physical Systems (CPS).

Irrespective of how we choose to count, it is clear that a new manufacturing era is upon us, and it is driven by information – a lot of information, more popularly known nowadays as ‘big data.’ In an opinion piece in a special issue of the Economist magazine, the chief executive of IBM argued that data is the ‘natural resource’ for the 21st century – just as steam power was for the 18th, electricity for the 19th, and hydrocarbons for the 20th [6]. She predicted that a new model of the firm would rise in 2014 using data as the natural resource, and called it the ‘smarter enterprise.’

While the private sector is preparing to exploit the digitization of manufacturing, many countries are investing in public-private partnerships to stimulate manufacturing innovation and get ahead in the new era. The United Kingdom has set up sixteen Centres for Innovative Manufacturing. They range from Additive Manufacturing to Ultra Precision, including Through-life Engineering Services. The German government, manufacturing industry, and academia are teaming up under the Industrie 4.0 umbrella, and are investing to preserve and advance their manufacturing leadership. Several Fraunhofer Institutes have demonstrated successfully the German model of public-private partnership to bring scientific ideas to industrial practice.

Since 2012, the United States of America has embarked on a major investment in a national network for manufacturing innovation [7], starting with six public-private partnership institutes. More such institutes are expected to join the national network soon. Several U.S. national research laboratories, including the National Institute of Standards and Technology (NIST), are investing in manufacturing-related research and development projects. In particular, NIST is investing in Smart Manufacturing, which is characterized by a heavy use of information, communication, and network technologies as befitting the needs of the new manufacturing era. Some of the enablers of the digitization of manufacturing, which in turn enables smart manufacturing systems, are described next.

3 Standards to enable smart manufacturing systems

Smart manufacturing systems require representations of engineering information that are entirely machine interpretable. However, the tradition of engineering drawings and textual documents that require human interpretation still dominate engineering practice throughout a product’s lifecycle. This practice is error prone and time consuming.

Smart manufacturing systems demand something better. They require engineering drawings to be replaced by augmented, digital three-dimensional (3D) geometric models; and, rich-text files of functional requirements, materials, processes, and maintenance to be replaced by structured information based on formal models. These replacements enable more manufacturing intelligence and automation based on fast and error-free processing and compositions of engineering information services (as directed by human) from the beginning to the end of a product's lifecycle.

Recent developments in standards provide some of the necessary tools and technologies to move towards machine readability. It is clear that no single software vendor or organization can cover the entire breadth and depth of a product's lifecycle. So, industry-developed standards have emerged as the natural choice to link disparate software systems and services. We describe some examples of the enabling standards in the following subsections.

3.1 Model-based 3D engineering

The International Organization for Standardization (ISO) has completed a major effort on a new standard, ISO 10303-242, titled 'Managed Model Based 3D Engineering.' It belongs to a family of standards informally called STEP (STandard for the Exchange of Product model data). ISO 10303-242 is also called STEP Application Protocol 242 (STEP AP242, for short). STEP AP242 combines many of the functionalities of its predecessors AP203 and AP214, and offers more [8]. Some of the new and improved functionalities in STEP AP242 that are of interest to through-life engineering support are described below.

Product Manufacturing Information (PMI) is a phrase used by the Computer Aided Design and Manufacturing (CAD/CAM) community to refer to Geometric Dimensioning and Tolerancing (GD&T), surface texture, finish requirements, process notes, material specifications, welding symbols, and other annotations. Some of this information is referred to as Geometrical Product Specifications (GPS), especially in the ISO parlance. PMI is also expanded as Product and Manufacturing Information, but the intent still remains the same. Fig 1 illustrates an example of presentation of PMI on a 3D model.

STEP AP242 PMI takes the first major step towards replacing two-dimensional engineering drawings with 3D models. More information on this new and exciting development can be found in [9]. It is interesting to note that the strongest business case for standardized PMI representation originally came from the LOTAR (LOng Term Archival and Retrieval) effort [10]. LOTAR, which is led by the aerospace industry, hopes to make the engineering information available in machine-readable form for long-term use of digital product and technical information.

Even though the need to archive 3D models with PMI was triggered initially by aerospace regulatory requirements, its appeal to all through-life engineering services goes well beyond the aerospace industry. In fact, any effort to remanufacture or reproduce anew a product or component during its use-phase requires PMI well after the component's initial manufacture. In particular, as we will describe in Section 7, several through-life engineering services require access to geometrical information about complex, long-lived products – preferably in 3D.

Recent developments in manufacturing technology have made archived 3D models even more valuable. For example, additive manufacturing, also known as 3D printing, has provided a greater ability to manufacture one-off or small-lot production of parts economically. But, to exploit the 3D printing technology for maintenance, repair, and overhaul, it is important to retain digital 3D models of parts – preferably in a neutral, standardized format – that endures technological changes for fifty years or longer.

The same pressure that drove industry to seek standardized PMI representation also pushed the development of a standardized representation of composite structures in STEP AP242. Fig 2 shows a part that has a complex composite-structure made up of plies and embedded electronic components. In addition to the final 3D structure, STEP AP242 composites representation must retain the lay-table information, capturing details of such composite layers and embedded components, that is critical for manufacturing. More details on the recent STEP composites capabilities can be found in [11].

As more products, especially in the aerospace sector, are manufactured using composite materials, their engineering information should be available in machine readable form for through-life engineering services. Repairing a damaged composite structure in service, for example, is no easy matter. It requires detailed information about the layers, ply orientations, and other embedded components to bring the damaged structure back to service quickly and correctly.

Another new capability in STEP AP242 is the Business Object Model (also known as the BO Model), which represents much of the standardized metadata associated with a product. BO model contains, for example, the assembly structure of a complex product. This assembly model, when combined with detailed 3D PMI for components of an assembly, provides a more complete set of computable information needed for through-life engineering services.

STEP AP242 contains a lot more capabilities than outlined above. It is important to emphasize that the capabilities described thus far are being implemented and tested in major CAD/CAM systems. The prospect for their wide-spread industrial adoption appears to be bright [9].

3.2 Models for Business objects

Most of ISO STEP AP242 described in Section 3.1 deals with geometry, except for the BO Model that deals with metadata associated with parts. Even these ‘business object models’ are closely tied to the assembly structure in which various parts are positioned spatially. This type of information is authored and stored in Product Data Management (PDM) systems that manage 3D CAD models.

But non-geometric data, such as bill of material (BOM), about a product are important for manufacturing and through-life engineering services. The BOMs are managed by Enterprise Resource Planning (ERP) systems and various other engineering information systems, such as Manufacturing Operations Management (MOM) systems and software systems that support maintenance and service. Some of these needs are addressed by Business Object

Documents (BODs), which are engineering and business message specifications developed by the Open Application Group Inc. (OAGi) [12]. The entire suite of specifications is called Open Application Group Integration Specification (OAGIS). OAGIS data models are created using the ISO 15000-5 Core Components Specification (CCS) as the modeling methodology [13]. OAGi released OAGIS Version 10 in 2013, the most current major release.

The high-level architecture of BODs is illustrated graphically in Fig 3. A BOD contains two areas: one devoted to application and the other to data. The Application Area contains information needed by the communicating infrastructure to deliver and track the message. It also contains context information that the receiver application may need in order to process the message correctly. Examples include the engineering or business process it is a part of, and whether it is a production or a test message. The Data Area contains the message content, which comprises Verbs and Nouns. The Verb indicates the action to be performed on the Nouns; the Noun conveys business-specific data to be acted upon by the receiver application.

Nouns are made up of reusable elements including Components and Fields. Components convey business data that have a complex structure. A component is made up of fields; and it may in turn contain other components. Fields convey business data that have a simple structure; i.e., a single value. Each Field is bound to a Data Type or a Code List that restricts its value domain. An important feature of OAGIS is its extension capability. OAGIS has a built-in extension capability for every component including the application area (not illustrated explicitly in Fig. 3).

OAGIS has recently adopted a model-driven approach (MDA), which separates the models from language-specific implementations. Fig 4 illustrates MDA realization in OAGIS 10 from OAGIS Model to OAGIS Expressions (language-specific implementations). The figure shows the packaging structure of OAGIS content. This results in two benefits to OAGIS users. First, OAGIS 10 defines the OAGIS Model, which is then derived into three OAGIS Expressions that are optimized for various deployment environments. For example, OAGIS JSON (JavaScript Object Notation) allows light-weight messages optimized for cloud and mobile deployments. This is an important development because, until recently, OAGIS focused exclusively on XML. Second, OAGIS Model packages reusable content into the Platform package. The package also includes BODs and Nouns that are agnostic to business and engineering domains. The Platform package allows for more pervasive adoption of OAGIS through other consortia, leading to greater interoperability for those using OAGIS.

Although BODs begin with the word ‘business’, they also support transactions in various engineering functional areas throughout a single enterprise or across multiple enterprises. Example transactions include design, manufacturing, supply chain, finance, sales, and accounting. OAGIS support for manufacturing integration is also extended by the Business-to-Manufacturing Markup Language (B2MML) message standard [14] published by MESA International. Recently, OAGi set up a smart manufacturing working group to push the OAGIS envelop further. This group will investigate how a reference model may be used to improve the reuse of information services relevant to manufacturing. OAGi has also started a

Semantic Refinement working group to improve the precision of OAGIS-based services in declaring their interface capabilities.

The breadth of verbs and nouns coverage, software vendor supports, and model-driven approaches have all enabled OAGIS BODs to support the composition of engineering services distributed over the cloud. In a recent OAGIS implementation case study, Fraunhofer Institute used OAGIS to develop a cloud solution for logistics [15]. The case study demonstrated the use of OAGIS to support compositions of logistics-software services in a cloud-computing marketplace environment.

3.3 Model-based systems engineering

Long-lived, complex artifacts are designed, built, and serviced using systems engineering principles. Thus far the requirements, realization, and maintenance of such systems have been managed largely using documents that are only human readable. Model-based systems engineering (MBSE) tries to change this practice by using machine-readable models instead of these traditional rich-text documents [16]. SysML is a standardized systems modeling language to enable MBSE [17].

SysML is an extension of the Unified Modeling Language (UML). Fig. 5 shows the relationship between UML and SysML using a simple Venn diagram. SysML defines additional diagrams, which are not contained in UML. These diagrams capture requirements and parameters that enable engineers to represent complex requirements, and to link them to systems simulation and analysis programs such as SIMULINK (including SimEvents for discrete event simulation of manufacturing systems) and MODELICA. The most recent version of SysML, Version 1.3 released June 2012, has been implemented by several leading systems engineering software vendors.

SysML is closely related to ISO 10303-233 (also known as STEP AP233) [18], which deals with systems engineering. STEP AP233, in turn, has several common features with ISO 10303-239 (commonly known as STEP PLCS) [19] that deal with product life-cycle support, which is of considerable interest to through-life engineering services. Therefore, we turn to that next.

3.4 Product life cycle support

Product Life Cycle Support (PLCS) [20] is the domain of ISO 10303-239, informally known as STEP AP239 (Standard for the Exchange of Product model data Application Protocol 239). Reference to STEP AP239 is often interchangeable with *PLCS*. At the minimum, PLCS provides standardized representations for product configurations during various phases of a product's lifecycle (e.g., as-designed, as-built, and as-maintained). But, it provides much more. PLCS also deals with in-service support requirements and related resources such as maintenance plans, schedules, job cards, and work request/orders. PLCS has a strong connection to model-based systems engineering described in Section 3.3. In fact, recent versions of PLCS are defined using UML/SysML.

The relationship between STEP AP233 and PLCS is shown in Fig. 6, where some of the functionalities of these two standards are also outlined. It is clear that these two standards share considerable features.

The defense sector has shown interest in implementing PLCS, where short- and long-term sustainment of weapon systems is paramount. Several pilot implementations of PLCS by major defense contractors are underway [21]. Section 7 outlines a scenario in a generic manufacturing plant floor (i.e., not restricted to the defense sector) that can benefit from the PLCS capabilities.

3.5 MTConnect

Smart manufacturing systems need coordinated intelligence across machines and devices. However, machines and devices are typically designed to function independently. Consequently, a standard is needed to build coordinated intelligence across diverse machines and devices. MTConnect is an open standard [22] to enable coordinated intelligence to be built on top of existing machines. Applications of MTConnect have enabled more efficient manufacturing production and through-life engineering processes by providing machine-readable data to intelligent applications.

MTConnect, developed by the MTConnect Institute, is for networking manufacturing devices and applications. It allows device data including subcomponents information, measurements, and events to be uniformly communicated to applications such as Manufacturing Operation Management (MOM), Performance Diagnosis and Prognosis (PDP), and predictive (e.g., condition-based) maintenance. The MTConnect standard is relatively easy to use because it relies on the popular HyperText Transfer Protocol (HTTP) and XML standards to deliver data.

Fig. 7a shows the types of data that may be provided by an *MTConnect* device. Fig. 7b shows a hierarchical machine structure and available data (in DataItem); the Component and DataItem can be cascaded into multiple levels in a hierarchy as necessary. The DataItem specifically describes the Streams (in Fig. 7a) available to the client. The Streams is a set of Samples, Events, or Condition for components or devices.

The Samples is a set of measurement values (e.g., temperature, spindle position) at a time point determined by a measurement frequency. The Events are discrete changes in a device's state, while the Condition indicates health and ability of a device to function such as Normal, Warning, Fault, or Unavailable. Multiple Faults and Warnings may be reported for a single data item, while only a single value can be reported for Samples and Events. Assets are mobile equipment that can be moved from one device to another such as cutting tools and fixtures.

MTConnect defines four services for clients to retrieve the data: probe, current, sample, and asset. The 'probe' service provides the Devices data. The 'current' and 'sample' services provide the most recent reading and a time-window-based Streams data, respectively. The 'asset' service provides Assets data. These services along with the basic concepts shown in

Fig. 7 provide a basis for more types of components, data items, and assets to be added to and communicate through the standard [22].

These increasingly available data exchange standards provide more opportunities for engineering information services to be composed and enable smarter manufacturing. However, industry still faces challenges in adopting some of these standards as described in [23] citing their complexity and overlapping capabilities. For standards that are deployed successfully, the trend is to exploit them using the service-oriented architecture (SOA), which provides convenient methods to compose software functions as described in the next section.

4 Service-oriented architecture

Standards, such as those described in Section 3, have been useful in enabling interoperability among disparate engineering software systems. The trend is to regard the functionalities provided by these software systems as services. This trend has accelerated recently with the arrival of cloud computing, which virtualizes computing and communication resources.

Service-oriented architecture (SOA) aims to achieve a distributed, loosely-coupled systems, such as a cloud-based-offering of software components discussed in the next section. In the service-oriented paradigm, software components are viewed as providing functionalities through services, which can be very specialized, creating greater heterogeneity. Services are virtualizations of software components. That is, service consumers do not need to know how service providers offer their services – from where, by which, or by how many software instances.

The service-oriented paradigm emphasizes visibility and semantics that enable (1) the matching between needs and capabilities, and (2) the composition of capabilities to address those needs. The visibility and semantics, in turn, are enabled by service descriptions and service contracts that capture the functional and non-functional information the service consumers and providers need to be aware of and agree upon.

SOA is commonly implemented using Web Services, which refer to a suite of standards from multiple standards development organizations; these standards include Web Service Description Language (WSDL) [24] and Business Process Execution Language (BPEL) [25]. However, such services may also be implemented using other strategies. Recently, SOA implementation using Representational State Transfer (REST), also known as RESTful Web Services, has gained widespread acceptance [26]. OASIS Open Data protocol (ODATA) [27] is an emerging standard for RESTful Web Services. The RESTful implementation is regarded as simpler and easier to use than the WSDL-based counterpart.

Smart manufacturing system architecture needs to enable agile response to changing needs in the manufacturing operations. To meet this requirement, cloud computing coupled with the service-oriented architecture is emerging as the architectural choice – creating a new opportunity for the next generation manufacturing systems. We explore the cloud computing promise and current capabilities next.

5 Cloud computing adoption in manufacturing

Two types of cloud computing adoptions in manufacturing have been proposed in [28]. The first is the direct adoption of cloud-computing technologies, which includes only the virtualization of computing and communication resources. The other, known as cloud manufacturing, extends the virtualization to manufacturing resources (such as machine tools and factories) and offers those resources as multi-tenant services to the cloud service customer. In this paper, we discuss the adoption of the first type of cloud computing and focus on the composition of engineering information services.

While cloud computing may be seen as an evolution of the computing-resource sharing idea that started in the 1960s, it may be argued that the current decade marks a revolutionary development of this fundamental idea with far reaching consequences. Functionally, cloud computing adds a number of capabilities – such as self-service, metered usage, and open platform architecture – to the resource virtualization. This opens the door for dynamic, scalable, and automated provisioning of computing resources all the way from computing hardware to application software in a pay-as-you-go business model. Such ubiquitous access to computing resources at a low up-front cost has resulted in many new and existing monolithic software solutions being offered as smaller services. Some of these services can be sourced individually. Others can be composed from standard-based solutions provided by several vendors to achieve the needed objectives with lower cost and greater agility [29, 36].

When assessing cloud-computing solutions, however, manufacturing enterprises still need to start by considering their applications and determine an optimal architectural model for the required functionalities. To guide cloud adoption, we propose a cloud architecture model, which comprises three essential cloud-architectural aspects: Deployment Model, Management Model, and Service Model. These aspects are summarized as the Cloud Architecture Decision cube in Fig. 8 and discussed further below.

5.1 Cloud deployment model

A cloud deployment model determines the level of resource (hardware and software) sharing. Deployment models can be classified as private, public, community, or hybrid [30]. A primary criterion determining the correct deployment model is the risk an enterprise is willing to take with its data. Private cloud carries the least risk because there is no resource sharing with another enterprise. Public cloud carries the most risk because resources are shared publicly across enterprises. And, community cloud is in the middle because it has resources shared within a restricted set of enterprises. Conversely, public cloud provides the highest cost-saving opportunity and private cloud the least. A large manufacturing enterprise might use a community cloud when there is a need for private, quick set-up, and tight collaboration among a group of partners. For example, a separate community may be set up for a specific supply chain.

Hybrid cloud is not really a separate deployment model. It simply indicates a mix use of public and private clouds. Most manufacturing enterprises will have a hybrid cloud, where sensitive information stays on the private cloud, and less sensitive and sharable information goes to the public cloud.

5.2 Cloud management model

A cloud management model describes the accessibility of physical computing resources (hardware) by the manufacturing enterprise. The key decisions to be made when selecting a management model are whether the computing resources are on-premise or hosted remotely, and whether they are managed by in-house or by third party personnel. Since the latter decision primarily impacts physical security measures and hosted resources are typically managed by third party, we reduce the management model to either on-premise or hosted.

In the on-premise model, the manufacturing enterprise manages access to the physical computing resources, and computational resources (software) are accessible via a private, local area network. On the other hand, cloud providers manage access to the physical computing resources in the hosted model; and, all computational resources are accessed through a wide area network, which opens up higher possibility for security and performance issues. Hosted solutions cost relatively less than their on-premise counterparts due to the higher resource/cost-sharing opportunities.[3] However, security is still a challenge to resource-sharing in the public or hosted cloud despite some recent arguments and evidences that private, on-premise computing is more vulnerable due to insufficient, outdated resources and attention to security [31, 32]. Nevertheless, some deployments will necessarily be on-premise because of the networking speed, criticality, and customization constraints. For the manufacturing domain, off-line simulation, analytics, and reporting applications are good candidates for hosted deployment [32, 34].

5.3 Cloud service model

A cloud service model determines the level of control and responsibility a cloud consumer and a cloud provider have for a particular cloud service. A typical cloud-computing reference architecture divides the level of control/responsibility into three layers that map to the classic software stack notation representing three categories of cloud services – namely, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [35]. As an end user whose functional requirements may constantly change and may be very specific to its manufacturing domain, the choice of service models is very important. It impacts the flexibility (customizability) the manufacturing enterprise has on a particular functionality (or application).

The IaaS service model offers the highest level of control to the manufacturing enterprise as the cloud consumer, followed by PaaS and SaaS. For example, a manufacturing enterprise can choose an engineering software application using an IaaS, PaaS, or SaaS service model. A SaaS-based approach would most likely result in a multi-tenant application where 1) multiple cloud consumers share the same instance of the application and 2) the level of customization by a consumer of such application service will likely be limited to adjusting its look and feel and the ability to add or turn-off some of its data fields. If customizations, such as different process flows or object states, are needed, the application will need to be serviced via PaaS or IaaS model. While these two service models would allow the

---

[3] It should be noted that public cloud is only available in the hosted management model. However, private cloud may be hosted or on-premise. Hosted private cloud can still share some resources such as facility and maintenance personnel.

manufacturer to retain direct ownership of, and responsibility for, the applications' configurations and its data, they would be more costly to operate and manage (e.g., they would require the cloud consumer to manage platform components, define how to scale computing resources). Nevertheless, it is possible that an application is built with SOA and a multi-tenant business process management environment that offers high-degree of customizations even in the multitenant SaaS context. This would be a feature to look for in the SaaS selection. Cloud architecture issues are discussed in greater detail in the following section.

6 Open cloud architecture for composable smart manufacturing systems

Fig. 9 illustrates a high-level functional view of an open cloud architecture for a manufacturing enterprise. Such an open cloud architecture is supported by open standards as well as open sources, which co-exist in a symbiotic relationship [36]. In the rest of this section, we discuss the essential constituents and challenges in developing and implementing the open-cloud architecture.

6.1 IaaS/OS layer

The IaaS/OS (Operating System) layer is a virtualization of computing hardware, shown at the bottom-most layer of Fig. 9. The layer consists of functional components that enable the scalable provision of computing resources. In particular, the Software Defined Computing component includes virtualized compute, network, and storage resources. The Virtual Machine Monitor (i.e., Hypervisor) component manages the Host Operating System component and the Guest Operating System component that may be needed by the upper layers to run on the virtualized computing resources. The Infrastructure Management component manages the provision of, and monitors the performances of, the infrastructure resources.

The OpenStack community [37] plays an important role in providing an open architecture at this layer. That open source community has projects in the areas of Software Defined Computing and Cloud Infrastructure Management. The Software-Defined Computing component provides Application Programming Interfaces (APIs) for fundamental computing resources such as compute, storage, and network. The Cloud Infrastructure Management component provides APIs for managing these resources. A standardization of the IaaS API has also been recently completed in ISO 19831 – Cloud Infrastructure Management Interface (CIMI) [38]. The OpenStack framework supports several virtual machines including open sources such as VirtualBox [39] and KVM (Kernel-based Virtual Machine) [40].

While IaaS provides a scalable computing resource, the PaaS layer discussed in the next section plays a crucial role to enable the composition of services in the manufacturing domain.

6.2 PaaS/Middleware layer

The PaaS/Middleware layer plays the important role of connecting the IaaS/OS layer, manufacturing hardware (including industrial internet components), enterprise partners, and the SaaS/Application layer, as shown in Fig. 9. We will now describe various components of the PaaS/Middleware layer depicted in Fig. 9, moving from bottom to top.

6.2.1 Data representation languages and messaging services—The Data Representation Languages and Messaging Services components provide fundamental information sharing and communication capabilities. For many manufacturing enterprises, different messaging services and data representations will be needed in different environments. In particular, new standards, such as Open Data (OData) protocol [27], Message Queue Telemetry Transport (MQTT) [41], and Advanced Message Queuing Protocol (AMQP) [42] have been developed (1) to work with existing standards, such as the Web Services [43] and OPC Unified Architecture [44] and, (2) to support the connectivity from the device level to the enterprise level. In addition, the Resource Description Framework (RDF) [45] and JavaScript Object Notation (JSON) [46] have emerged to complement the Extensible Markup Language (XML) [47]. Together, they fill needs at different ends of the data representation spectrum, by seeking an optimal balance between semantic expressivity and clarity, on one end, and bandwidth concerns on the other.

6.2.2 Application development and deployment—The PaaS/Middleware layer also provides a platform for ‘massaging’ engineering data, which are shown in Fig. 9 inside a bigger box called Application Development and Deployment. The Data Analytics component is the key to providing a new kind of intelligence that responds to the smart manufacturing vision discussed in Section 2. For that, a significant amount of both real-time and event-based data coming from the shop floors, warehouses, sales, and supply-chain partners need to be cleaned, aggregated, contextualized, and analyzed. Semantic Mediation, Workflow (data-driven), SQL Database, and NoSQL[4] Database provide support for the Data Analytics component. There have been significant developments in several open-source communities to support these performance-hungry data-analytics’ functionalities, such as BigData [48] and Sesame [49] for Semantic Mediation, Taverna [50] for SOA-based data-driven Workflow, Apache Hadoop [51] for databases, and data processing APIs (application programming interfaces). The open source R [52] and the Predictive Model Markup Language (PMML) standard [53], on the other hand, support the portability of predictive models across tools for data-analytics.

The PaaS Deployment Management component is one of the key differentiators of the cloud-based, middleware layer from the non-cloud counterpart. It is responsible for optimized execution and auto-scaling of applications as cloud services. The deployment management software typically also provides value-added functionalities such as the application life-cycle management that automatically deploys differing PaaS technology components (e.g., run-time environment, application server, messaging, and database) required by differing applications. In this domain, for example, Cloud Foundry [54] is an open source PaaS project that is extensible to support a variety of PaaS-layer technological components.

---

[4] NoSQL is known as “no SQL” or “not only SQL”, referring to several classes of non-relational databases discussed later.

OASIS Cloud Application Management for Platforms (CAMP) [55] and Topology and Orchestration Specification for Cloud Applications (TOSCA) [56] are two standards advancing the portability of cloud applications across cloud platforms. CAMP, in particular, focuses on providing a service-oriented API specification for managing the application deployment life-cycle. CAMP also provides a specification for describing PaaS-layer technological components (or more generally resources) required by an application, independent of a specific platform's components. It assumes that the IaaS-layer components will be managed by the PaaS system. TOSCA, on the other hand, has a larger scope. It describes all the resources required by an application (also independent of specific platform's components) and also the processes needed to deploy and provision the whole application as well as manage its life-cycle. While these two standards may not ensure with absolute certainty the portability of an application across various cloud providers, an application developed with components based on these open standards is more likely to be portable.

6.2.3 PaaS marketplace and its management—Within a cloud-enabled service ecosystem, services performing identical functions may be offered by multiple service providers (internal as well as external to the enterprise), albeit with differing service levels. The PaaS Marketplace component allows PaaS consumers to source services from these specific providers to meet their particular requirements. TOSCA and CAMP also aim at enabling this open ecosystem of services. However, both of them only define a generic data structure and interface for describing the service's functional and non-functional characteristics. That is not enough. Service and function reference models are needed to provide common semantics for describing those characteristics. These characteristics, however, can be complex and evolving. The Marketplace Management component should provide functionalities to deal with these issues.

The Service Discovery & Composition component assists in managing the complexity of services compositions by providing multi-criteria decision support for the user to match requirements with capabilities. The Reference Model Management component assists the community with the evolution and other life-cycle-management aspects of the reference model. The same set of marketplace-management functionalities in the PaaS layer will also support the SaaS Marketplace. The diversity of manufacturing domains, where a functional characteristic may be specific to an industry or even product type, makes these functionalities even more important in the SaaS layer.

We believe that the marketplace is the area where significant opportunities for innovations present. At the same time, several hurdles need to be crossed to fully realize the open architecture. Next section discusses in more details this vision and hurdles as supported by several, recent publications discussed in the next section.

6.3 SaaS/Application layer

An on-going evolution of the cloud-based SaaS/Application layer is being fuelled by the increasing ability to quickly build up a new business value. This ability is achieved by composing new and existing software and service assets that are internally or externally operated. Industry terms this ability, “An API economy.” [36] To make the API economy a sustained capability, information and application functions need to be accessible as discrete services supported by the SaaS Marketplace. Such accessibility is a necessary condition for external services to be composed, and, therefore, able to deliver value to manufacturing enterprise efficiencies and qualities.

The challenges to supporting this ecosystem of internal and external services are the increasing needs of standards for data exchange such as those described in Section 3. These standards support the interoperation of exemplar categories of functions identified in the SaaS/Application layer and the information services that arise from the Manufacturing Hardware and Enterprise Partners.

While data exchange standards facilitate data-level interoperability, new standards to facilitate interoperability at the functional and non-functional levels are also needed. These new standards are needed to enable specifications of functional and non-functional requirements and capabilities for services that are shared across the ecosystem. This in turn would allow for effective search, discovery, composition, and configuration management of services. However, the recent standard roadmap published in an Industrie 4.0 report has indicated that standards in this area are underdeveloped [57]. Fortunately, there are both existing standards and research results that provide a basis to build upon.

For the functional specification, the NIST SIMA Reference Architecture Activity Models [58] provide an integrated, hierarchical view of manufacturing enterprise functions. ISA-95 and ISA-88 standards [59, 60] provide a basis for the Manufacturing Operation Management (MOM) Functions [61]. OAGIS provides a basis for the Enterprise Resource Planning (ERP) and Supply Chain Management (SCM) Functions [62, 63]. And, the PLM (Product Lifecycle Management) Services standard provides a basis for PLM and Digital Manufacturing (DM) Functions [64, 65]. The non-functional specification needs to take into account 1) existing standards such as security and communication protocols and 2) other yet-to-be-developed standards to address licensing model, computational performance, safety, and reliability characteristics of SaaS services [66].

The current status of the manufacturing standards in support of SaaS Marketplace can be compared with standards in other, more advanced industry domains. For example, the telecommunications industry has developed an extensive functional model that outlines a four-level-deep canonical decomposition of functions found in telecommunications enterprises [66]. Such a functional model is essential for rapidly designing and configuring systems by allowing effective specifications of customer requirements as well as providers’ service capabilities.

The Smart Manufacturing Working Group at the OAGi [12] is addressing these much needed functional specification standards for the manufacturing domain. The group is building upon existing standards including the Process classification framework [66], the Supply Chain Operation Reference (SCOR) model [68], and ISA-95.

Ongoing research initiatives around the world recognize the importance of standards for functional and non-functional requirements’ specifications to enable manufacturing services interoperability and composition [5, 69]. One direction of research investigates alternative methods of developing, adopting, and managing reference models and architectures. In that regard, discussions of the method for bottom-up development of reference models have been appearing recently in both industry and academia [5, 69]. This development may have a significant impact on the Reference Model Management component discussed earlier in the PaaS layer.

The preceding three sections, sections 4, 5, and 6, outlined the SOA and open cloud architecture. Although they provide the paradigm and technology platform to enable dynamic composition of engineering information services, they themselves are not a solution to domain-specific problems. The following section marries the engineering standards described in Section 3 with SOA and cloud computing to compose a domain-specific solution to problems in a through-life, engineering-service scenario.

7 Composing services

Any engineering information system can provide a service. An important question we should ask is whether such services can be composed to provide a bigger service that addresses a customer's need. In addition, we should ask how quickly such a composition can be put together or modified in a dynamic industrial and business environment. The best way to answer these questions is (1) to gather realistic customer problems that require such a composition of existing services, (2) to test the hypothesis that the services are composed correctly, and 3) to show that the resulting composition actually solves the customer problems. Consider the following scenario, formalized in the Business Process Model Notation (BPMN) [71] in Fig. 10, which describes a service call affecting a broken down production line in a manufacturing plant [72–74]. Shaded rectangular boxes in the figure are SaaS applications deployed in the hosted-cloud. Others are on-premise applications:

"A fault from an Electrical Control Unit (ECU) for the motors powering the plant's central conveyor line is detected by a Performance Diagnosis and Prognosis (PDP) system, which monitors and brokers all critical plant equipment over the plant's wireless local area network (WLAN). The equipment operator reviews information from the PDP and creates an 'ECU Fault Event' that is instantly dispatched to a Manufacturing Operations Management (MOM) system. This event is processed by the operations manager, who uses a Decision Support (DS) system hosted in the public cloud to determine if the fault is a false alarm or a new alarm. In performing this Analyze ECU Fault task (see Fig. 10), he has to interact with several systems. He has to manually access the motor's calibration and instrument reading in the PDP and SCADA (Supervisory Control and Data Acquisition) systems hosted on-premise to confirm that the fault is real. He also has to instruct the PDP to compare the ECU's signal history with the manufacturer's specifications to find out whether the ECU is likely to fail soon.

"The operations manager then prepares an audit report of the ECU's previous maintenance and performance within the MOM system. In this Create ECU Audit Report task, he additionally accesses the ERP (Enterprise Resource Planning) system to investigate the availability of a warehoused, replacement ECU. Because of the shortage, he makes a direct query to ECU manufacturers and finds an equivalent part from another vendor. The ERP system also shows a field performance upgrade that improves the current ECU's operational characteristics.

"The operations manager attaches the ECU audit report along with a high-priority work order he creates in MOM. MOM automatically issues an alert to an on-duty field technician to the problem via a Smart Phone message. In the Review Work Order task, the technician uses his (mobile) tablet to identify the ECU's unit number, physical location, safety notifications, and a brief description of the fault type from the work order.

"In the Repair ECU task, the technician locates the faulting ECU and takes the motor off-line by locking out its power system—a standard safety procedure appearing in his tablet's checklist. The technician also uses his tablet to access the PLM SaaS system to retrieve design specification, installation, configuration, and testing procedures. He notices an optional, performance-enhancement service bulletin and compares output signals with the failing ECU. After consulting with Operations, he applies the optional upgrade to bring the conveyor motor back into a no-fault operating condition.

"The technician downloads the performance package to the ECU along with the vendor's recommended testing and startup procedures. The installation and pretest are quickly completed. He refers to the standard restart procedures and brings the conveyor back on line. Then, the ECU is monitored locally to ensure that the startup sequence has not stressed the ECU or motor beyond performance standards.

"The technician gathers the ECU's operational history and diagnostic outputs from the PDP and maintenance history from the PLM in the Create and Send the ECU Failure Report tasks. The ECU's manufacturer will investigate the circumstances to determine if there is a fundamental design flaw in the Manage Product Failure Report task. The service request is closed out and standard operations resume."

The scenario described above illustrates several key ideas. First, humans play important supervisory and collaborative technical roles; but, humans should not be required to re-enter information that already resides in a trusted source. Also, human intervention should not be required to read and interpret textual or graphical information. That information should be represented in a machine-readable form that can be interpreted quickly and correctly by a computer. These are some of the key elements of smart manufacturing.

In a realization of smart manufacturing, the reader can envision the automation of several complex tasks that are manually carried out in the scenario outlined above. Figs. 10 and 11 illustrate this point using BPMN diagrams for the ECU service call process. Notice that manual user tasks (each notated by a box with a user icon) in Fig. 10 including the Analyze ECU Fault, Create ECU Report, and Create Work Order have become automated service tasks (each notated by a box with a gear icon) in Fig. 11. In addition, two user tasks including the Repair ECU and Create ECU Failure Report in Fig. 10 have become partially-automated composite tasks (each notated by a box with the plus sign) in Fig. 11. These automations eliminate the need for a human to re-collect and re-enter data from several systems. This makes the process faster and reduces errors.

Implementing even this relatively simple scenario as a valuable composite process involves several on-premise and cloud-hosted engineering information systems. Several services, providing different functions, from several systems, need to be found and composed. It is highly unlikely that one single software vendor or organization will provide all the services and compose them using its own proprietary data and interfaces.

One solution to such a dynamic composition problem is to use standardized data, such as those described in Section 3, and standardized service interfaces using SOA and cloud computing as described in Sections 4 to 6. As the scenario indicates, such data can come from heterogeneous systems including PLM, ERP, MOM, CRM, PDP, and SCADA. Considerable data analytics are also employed to monitor equipment health, diagnose problems, and suggest corrective actions.

These data and systems are now moving to clouds, and the scenarios are changing fast to reflect demanding business needs. Hence, we face the urgent need for dynamically composing these engineering information services in the cloud. Standards are necessary to address this heterogeneity and to increase the agility of operations. As illustrated in Fig. 11, while existing standards such as the OMG PLM Services and the OAGIS BOM BOD are available to support compositions of engineering information services in the scenario, several new standards are needed, such as Corrective Action and Preventive Action (CAPA) messages and Device History messages among others. Therefore, there is also an urgent need for additional manufacturing-related standards. Standards-based data security and authentication mechanisms are also needed to reduce complexity in connecting on-premise and hosted services.

8 Summary and concluding remarks

In the current era of digitization of manufacturing to enable smart manufacturing, engineering information systems play a central role. No single software vendor or organization can provide all the software and services necessary to run an efficient enterprise. Therefore, proprietary data and interfaces are not a viable option to serve the lifecycle and supply networks of products that have to live through years of technological evolutions. Hence, standards have to assume an important role.

In this paper, we described some of the standards that have been created, or upgraded recently, to meet the demands of smart manufacturing. These standards are based on information models with representations that are machine-readable. By avoiding human interpretations and interventions as much as possible, costly and time-consuming errors can be avoided.

With the aid of service-oriented architecture, these standards also enable composition of engineering information services to meet more complex and fast-changing engineering and business needs. We have already seen some successes in deploying such services in industry; but, many challenges remain before we can realize the full potential.

Cloud-based engineering information services are still in their infancy. Breaking the proprietary hold on data and interfaces still remains a problem in manufacturing. As open standards for data and interfaces become more popular, innovative entrepreneurs will use them to open up new markets for services. Effective search, discovery, composition, and configuration management of manufacturing services supported by standards for functional and non-functional specifications will be needed. This is especially important for small- and medium-sized companies who cannot afford costly monolithic solutions. We need more software technologies and tools that use open standards to define and compose the engineering information services in manufacturing, which places a higher premium on timeliness and reliability. So, we also need a better communication infrastructure (e.g., more deterministic Ethernet), and better cyber security for both wired and wireless communication.

Acknowledgments

This paper is based on a keynote address delivered at the 3rd International Through-life Engineering Services conference held at Cranfield University, U.K. during Nov. 4–5, 2014. We thank our colleagues in several standards development organizations involved in developing and upgrading the standards described in this paper. We are especially grateful to Dr. Albert Jones (NIST), Dr. Jacques Durand (Fujitsu Software, US), and Sharon Kemmerer (NIST) for their timely reviews and comments on several drafts of this paper.

References

1. McMahon, C.; Ball, A. Information systems challenges for through-life engineering. 2nd International through-life engineering services conference, Procedia CIRP; 2013. p. 1-7.
2. Srinivasan V, Lämmer L, Vettermann S. On architecting and implementing a product information sharing service. ASME J of Computing and Information Science in Engineering. 2008; 8(1)
3. Srinivasan V. An integration framework for product lifecycle management. J of CAD. 2011; 43(5): 464–478.
4. The Economist. April 21 issue on the Third Industrial Revolution. 2012.
5. Securing the future of German manufacturing industry. Acatech; Germany: 2013. Recommendations for implementing the strategic initiative INDUSTRIE 4.0.
6. Rometty, V. The Economist, The world in 2014 edition. 2013. The year of the smarter enterprise.
7. Advanced manufacturing portal. http://manufacturing.gov
8. ISO 10303-242:2014, Industrial automation systems and integration -- Product data representation and exchange -- Part 242: Application protocol: Managed model-based 3D engineering.
9. Barnard Feeney, A.; Freschette, S.; Srinivasan, V. A portrait of an ISO STEP tolerancing standard as an enabler of smart manufacturing systems. 13th CIRP Computer Aided Tolerancing Conference; Hangzhou, China. 2014.
10. Long term archival and retrieval. http://www.lotar-international.org
11. Hunten K, Barnard Feeney A, Srinivasan V. Recent advances in sharing standardized STEP composite structure design and manufacturing information. J of CAD. 2013; 45(10):1215–1221.
12. Open Application Group Inc. (OAGi). OAGi Integration Specification Release 10.1. 2014.
13. ISO 15000-5. Electronic business extensible markup language (ebXML) – Part 5: Core components specification (CCS). Geneva, Switzerland: 2014.
14. MESA International. Business to manufacturing markup language V0401. 2008.
15. Weibenberg N, Springer N. Cloud process modeling for the logistics mall-object-aware BPM for Domain Experts. Open J of Mobile Computing and cloud Computing. 2014; 1:31–49.
16. Model Based Systems Engineering Wiki. www.incosewiki.org.uk/Model_Based_Systems_Engineering/index.php?title=Main_Page
17. Object Management Group. Systems Modeling Language v1.3. 2012.
18. ISO 10303-233. Industrial automation systems and integration – Product data representatin and exchange – Part 233: Application protocol: Systems engineering. Geneva, Switzerland: 2012.

19. ISO 10303-239. Industrial automation systems and integration – Product data representation and exchange – Part 239: Application protocol: Product life cycle support. Geneva, Switzewrland: 2012.
20. Product life cycle support. plcs-resources.org
21. Perkins, H. Exploiting PLCS within UK MOD. 2003. www.oasisopen.org/committees/download.php/5434/10.%20Exploiting%20PLCS%20within%20UK%20MoD%20-%20Howard%20Perkins.pdf
22. MTConnect Institute. MTConnect standard version 1.3.0. 2014.
23. Ivezic, N., et al. NIST Technical Note (NISTIR 8124). 2016. OAGi/NIST Workshop on Open Cloud Architecture for Smart Manufacturing.
24. W3C. Web Services Description Language V. 2 Part 0: Primer. 2007.
25. OASIS. Web Services Business Process Execution Language V 2.0: Primer. 2007.
26. Representaional State Transfer Web Services: The basics. www.ibm.com/developerworks/webservices/library/ws-restful
27. OASIS. Open Data Protocol Version 4.0. 2014.
28. Xu X. From cloud computing to cloud manufacturing. Robotics and computer-integrated manufacturing. 2012; 28(1):75–86.
29. Oracle White Paper in Enterprise Architecture – Achieving the Cloud Computing Vision. Oct.2010
30. Mell P, Grance T. The NIST definition of cloud computing. 2011 NIST SP 800-145.
31. Linthicum, D. The public cloud is more secure than your data center. Info World. Dec. 2015 Internet Web Article. http://www.infoworld.com/article/3010006/data-security/sorry-it-the-public-cloud-is-more-secure-than-your-data-center.html
32. Krazit, T. Are public clouds really safer than private data centers?. Oct. 2015
33. Invensys Systems, Inc. and Microsoft Corporation. The cloud for manufacturing. Fortune. 2014. Internet Web Article. http://fortune.com/2015/10/20/structure-2015-public-cloud/
34. Doherty, F.; Moret, B. Tomorrow: The implicaitons of the cloud in manufacturing, Rockwell Automation Presentation Slides. 2012.
35. Liu F, et al. NIST Cloud computing reference architecture. 2011 NIST SP 500-292.
36. Diaz, A.; Ferris, C. IBM's Open cloud architecture. 2013.
37. OpenStack. www.openstack.org
38. ISO/IEC 19831 Cloud Infrastructure management Interface (CIMI) Model and RESTful HTTP-based Protocol – An Interface for Managing Cloud Infrastructure.
39. VirtualBox. www.virtualbox.org
40. Kernel Based Virtual Machine. www.linux-kvm.org
41. OASIS. Message Queue Telemetry Transport V. 3.1.1. 2014.
42. ISO/IEC 19464. Advanced Message Queuing Protocol. 2014.
43. W3C Web Services. www.w3.org/2002/ws/Activity
44. OPC Foundataion. OPC Unified Architecture Specification V. 1.0.
45. W3C. Concepts and Abstract Syntax. 2014. Resource Description Framework 1.1.
46. ECMA International. The Javascript Object Notation (JSON) data interchange standard. 1. 2013. ECMA-404
47. W3C. Extensible Markup Language 1.0. 1998.
48. BigData Graph Database. sourceforge.net/projects/bigdata
49. Sesame RDF processing. rdf4j.org
50. Tan W, et al. A Comparison of Using Taverna and BPEL in Building Scientific Workflows: the case of caGrid. Concurrent Computing. 2010; 22(19):1098–111.
51. Apache Hadoop. hadoop.apache.org
52. The R Project for Statistical Computing. www.r-project.org
53. Data Mining Group. Predictive Model Markup Language V. 4.2.1. 2014.
54. Cloud Foundry. www.cloudfoundry.org
55. OASIS. Cloud Application Management for Platforms V. 1.1. 2014.

56. OASIS. Topology and Orchestration Specification for Cloud Applications V. 1.0. 2013.
57. VDE Association for electrical, electronic & information technologies. The German standardization roadmap, Industrie 4.0. Version 1.0. 2014.
58. Barkmeyer, E., et al. SIMA Reference Architecture, Part 1: Activity Models, NISTIR 5939. 1999.
59. ANSI/ISA-95.00.01-2000, Enterprise-Control System Integration Part 1: Models and Terminology.
60. ANSI/ISA-88.00.01-2010 Batch Control Part 1: Models and Terminology.
61. Younus, Muhammad, et al. MES development and significant applications in manufacturing-A review. Education Technology and Computer (ICETC), 2010 2nd International Conference on; IEEE; 2010.
62. Murray, M. Material Management with SAP ERP. 3. 2011.
63. Gerald, B., et al. Oracle E-Business Suite Manufacturing & Supply Chain Management. 2001.
64. CIMdata. A CIMdata White Paper. 2010. Teamcenter “unified” “Siemens PLM Software’s Next Generation PLM Platform”.
65. CIMdata. A CIMdata Report. 2011. Digital Manufacturing - “Enabling lean for more flexible manufacturing”.
66. CISCO. White paper. 2009. Introduction to eTOM.
67. American Productivity and Quality Center. Process classification framework V. 6.1.1. 2014.
68. Supply Chain Council. Supply Chain Operation Reference Model Revision 11.0. 2012.
69. European Commission. Factories of the Future: Multi-annual roadmap for the contractual PPP under Horizon 2020. 2013.
70. Microsoft Corporation. Microsoft’s Upstream Reference Architecture (MURA). 2013.
71. Object Management Group. Business Model and Notation. V. 2.0. 2011.
72. Srinivasan, V. Sustaining manufacturing assets through smarter utilization of information and communication technologies. 5th Annual IEEE Conference on Automation Science and Engineering; Bangalore, India. Aug. 22–25; 2009. p. 478-482.
73. Popko, E.; Luyer, E. IBM White paper. 2009. Asset management to support product lifecycle management (PLM) – leveraging asset management to benefit PLM projects.
74. Ouertani, ZM.; Srinivasan, V.; Parlikad, AK.; Luyer, E.; McFarlane, D. Through-life active asset configuration management. 6th International Conference on Product Lifecycle Management; Bath, U.K: University of Bath; 2009.

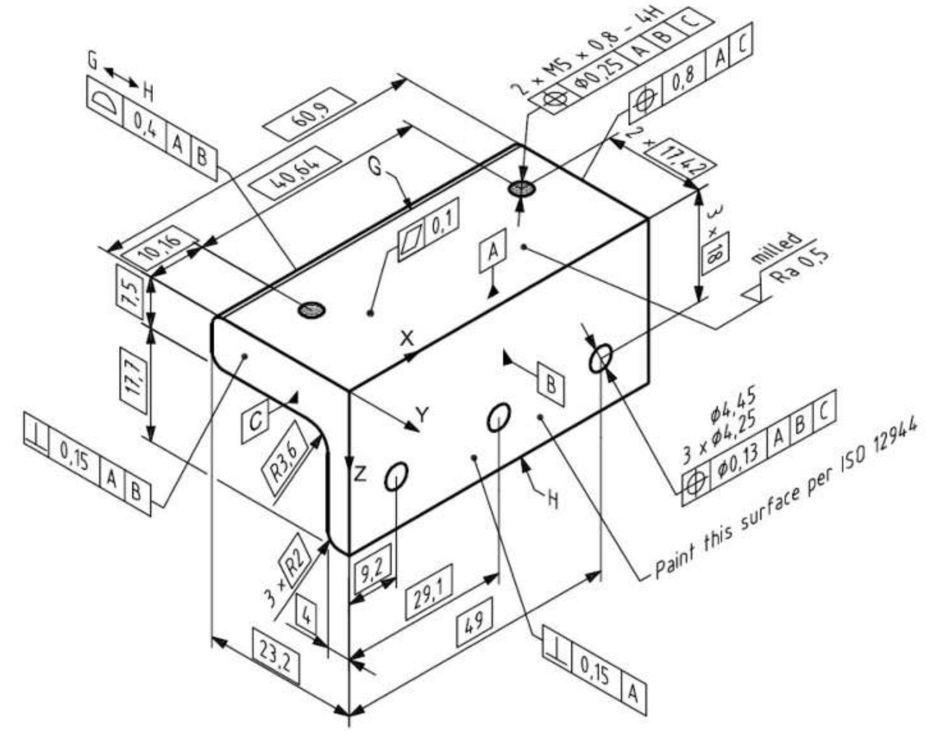

Fig. 1.

An example of standardized presentation of product manufacturing information (PMI) on a 3D model

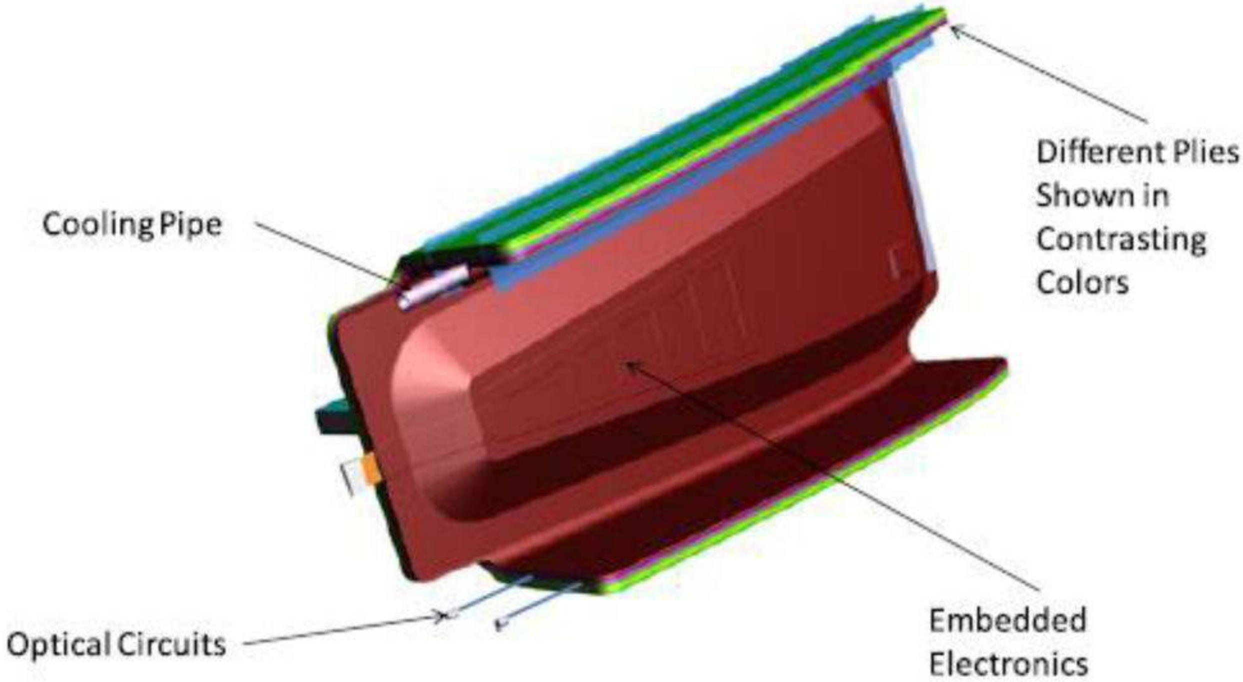

Fig. 2.
An example of complex, composite structure representable by STEP AP242

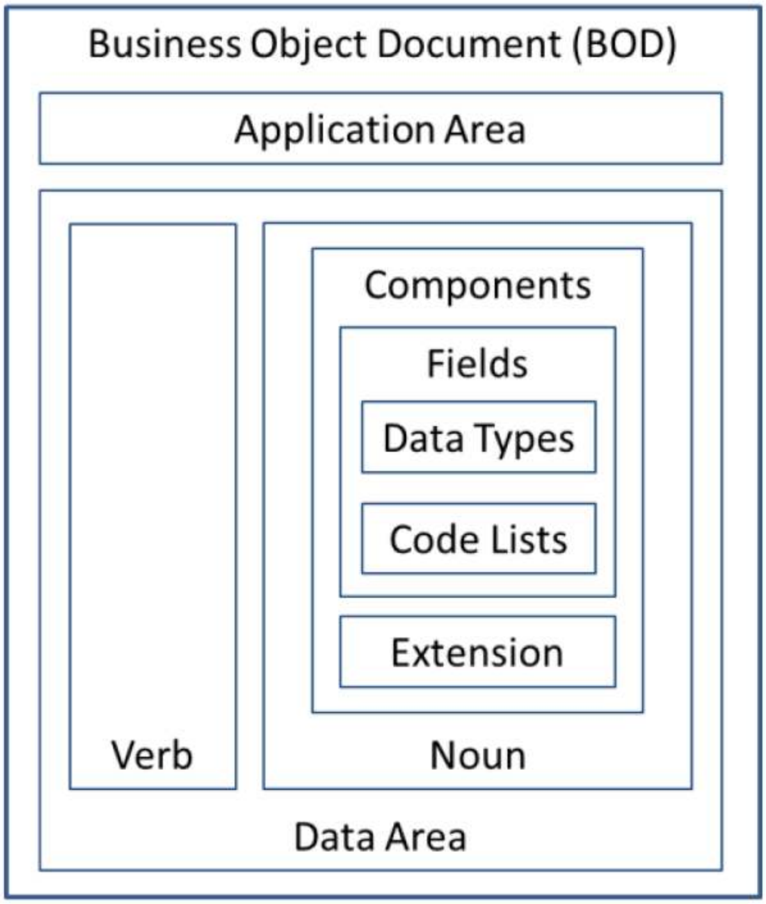

Fig. 3.
High-level business object document (BOD) architecture

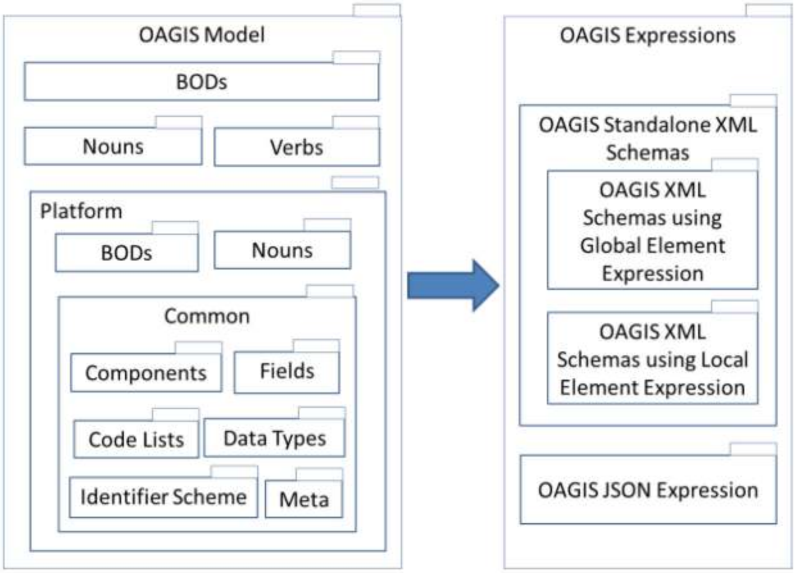

Fig. 4.
OAGIS MDA realization and delivery structure

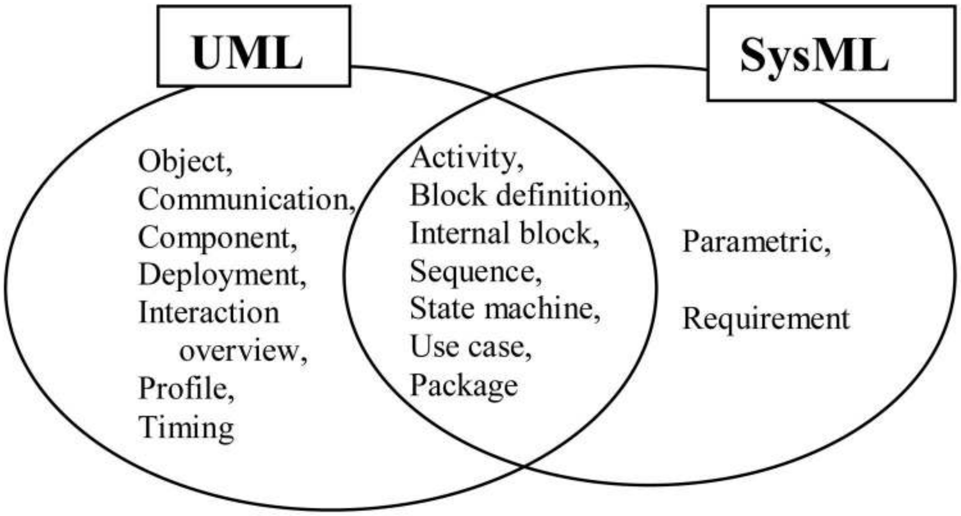

Fig. 5.
Relationship between UML and SysML diagrams

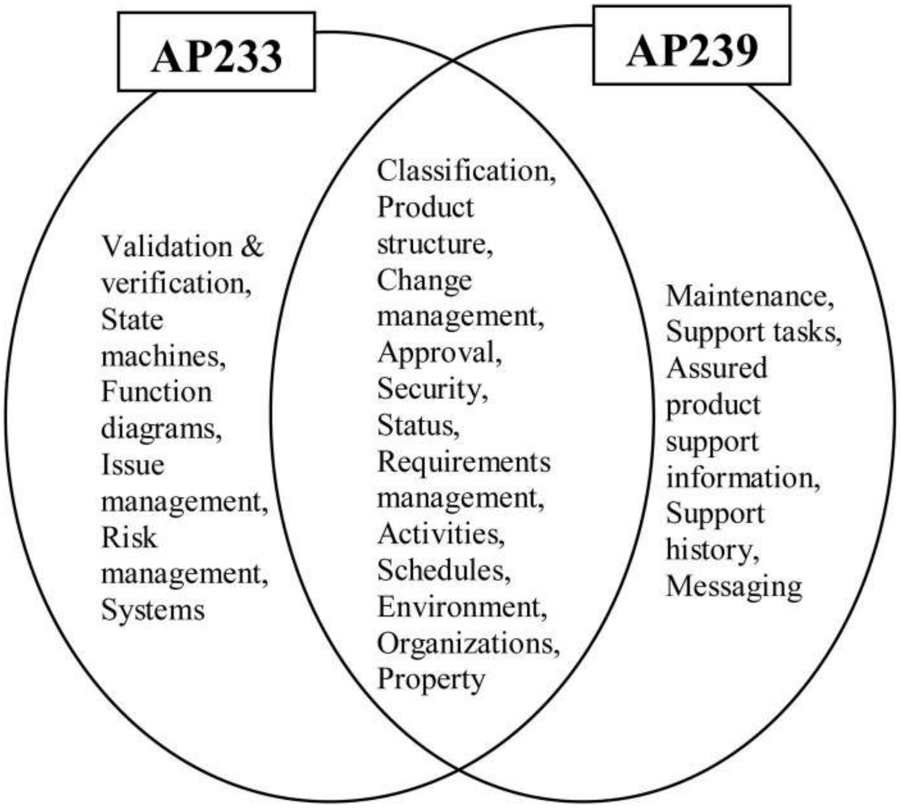

Fig. 6.

Relationship between ISO STEP AP233 and AP239 (PLCS)

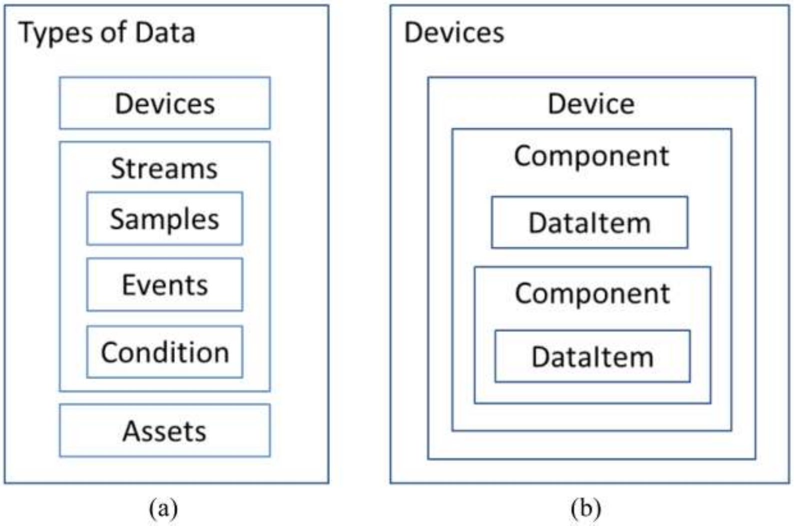

Fig. 7.

(a) Types of data in MT Connect; (b) Device data structure

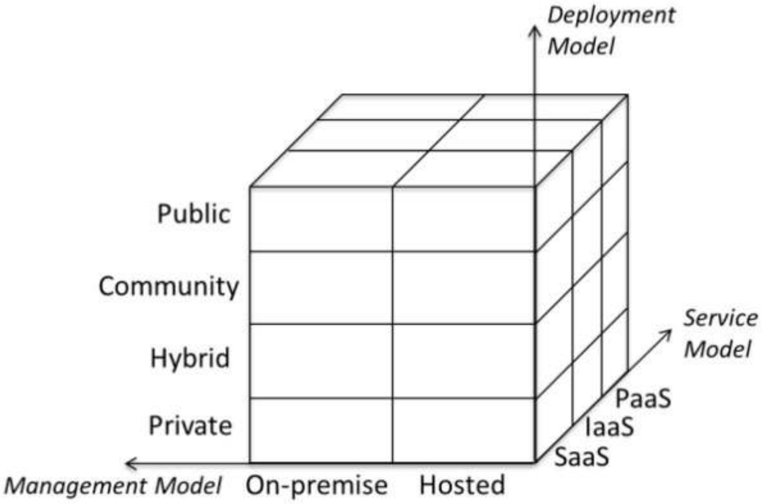

Fig. 8.
Cloud architecture decision cube

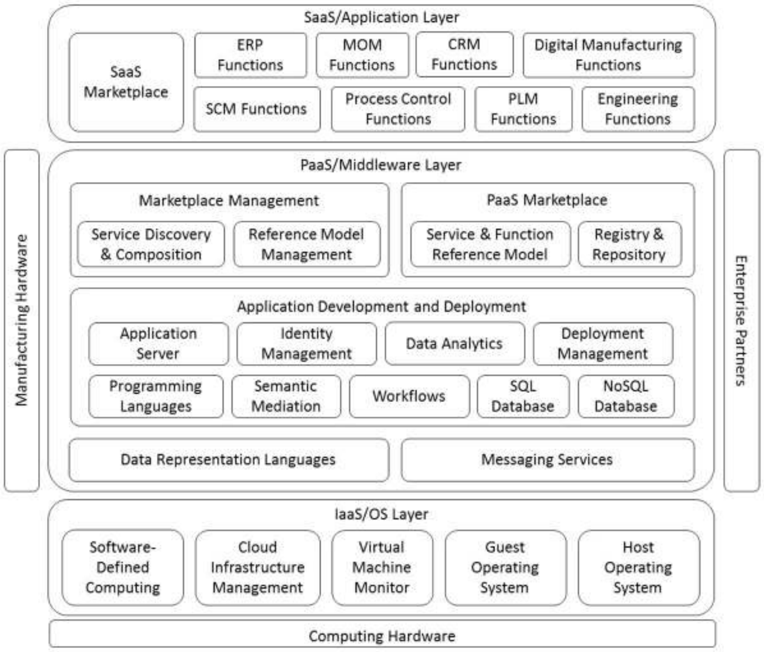

Fig. 9.

Open cloud architecture for composable smart manufacturing systems

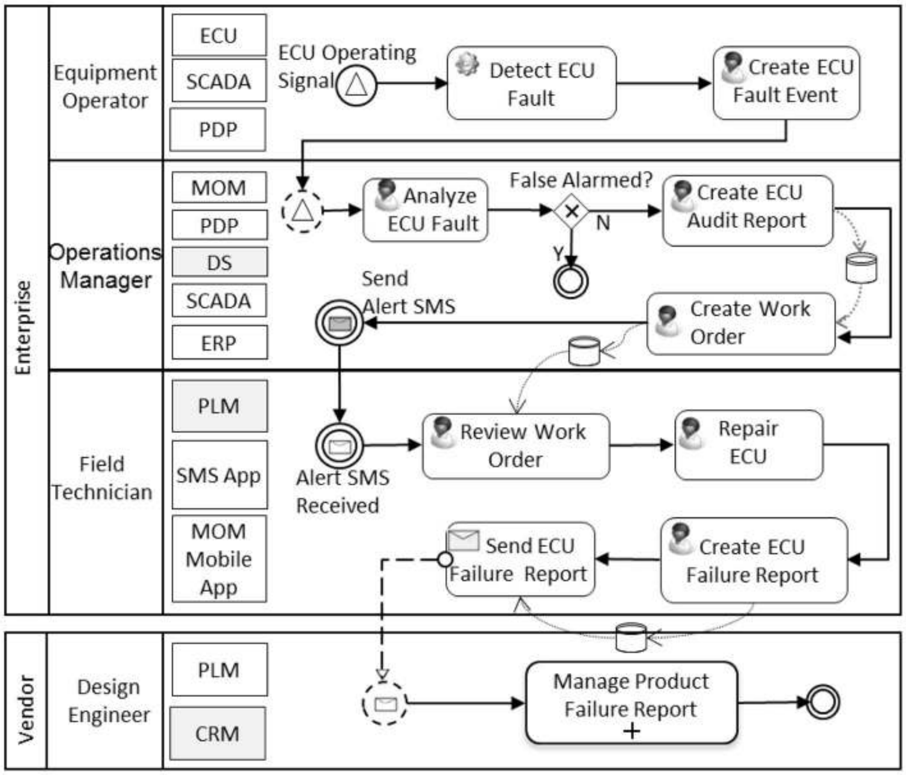

Fig. 10.

As-is ECU service call process

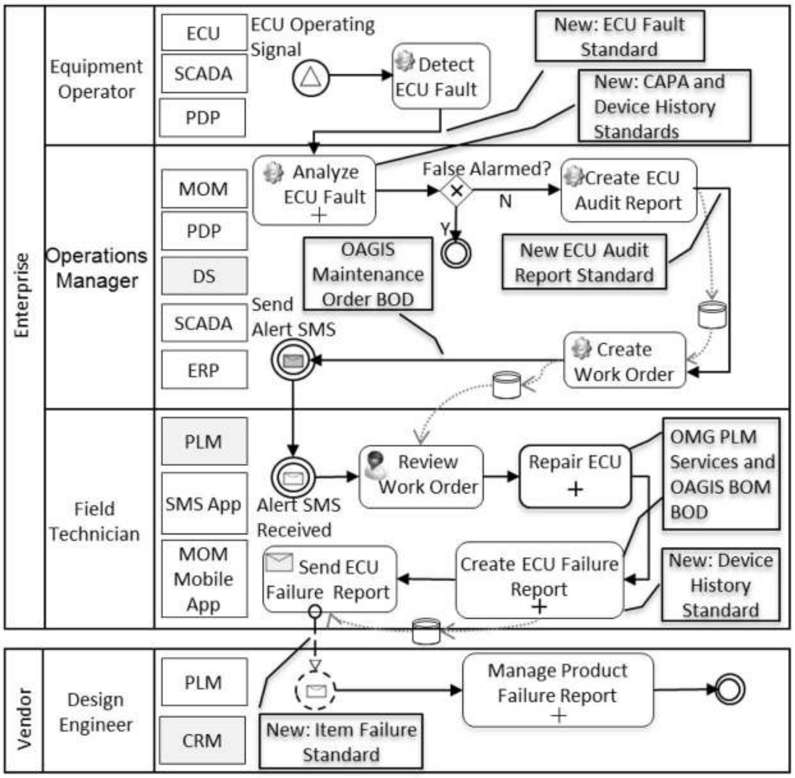

Fig. 11.

To-be ECU service call process