The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-018-2416-9

ORIGINAL ARTICLE

Developing a big data analytics platform for manufacturing systems: architecture, method, and implementation

Jungyub Woo [1] · Seung-Jun Shin [2] · Wonchul Seo [2] · Prita Meilanitasari [2]

Received: 14 March 2017 / Accepted: 11 July 2018

Abstract

Manufacturing industries have recently promoted smart manufacturing (SM) for achieving intelligence, connectedness, and responsiveness of manufacturing objects consisting of man, machine, and material. Traditional manufacturing platforms, which identify generic frameworks where common functionalities are shareable and diverse applications are workable, mainly focused on remote collaboration, distributed control, and data integration; however, they are limited to incorporating those characteristic achievements. The present work introduces an SM-toward manufacturing platform. The proposed platform incorporates the capabilities of (1) virtualization of manufacturing objects for their autonomy and cooperation, (2) processing of real and various manufacturing data for mediating physical and virtual objects, and (3) data-driven decision-making for predictive planning on those objects. For such capabilities, the proposed platform advances the framework of Holonic Manufacturing Systems with the use of agent technology. It integrates a distributed data warehouse to encompass data specification, storage, processing, and retrieval. It applies a data analytics approach to create empirical decision-making models based on real and historical data. Furthermore, it uses open and standardized data interfaces to embody interoperable data exchange across shop floors and manufacturing applications. We present the architecture and technical methods for implementing the proposed platform. We also present a prototype implementation to demonstrate the feasibility and effectiveness of the platform in energy-efficient machining.

Keywords Big data analytics · Holonic manufacturing system · Cyber-physical system · Agent system · Energy efficiency · Predictive modeling

1 Introduction

The future ability of manufacturing industries depends on the smartness of manufacturing systems; smart manufacturing (SM) will make it possible to achieve higher agility, productivity, and sustainability [1, 2]. The Smart Manufacturing Leadership Coalition defines SM as "the intensified application of advanced intelligence systems to enable rapid manufacturing of new products, dynamic response to product demand, and real-time optimization of manufacturing production and the supply chain network [3]." Although there are several definitions of SM apart from the definition above, one remarkable and common understanding is that manufacturing systems should evolve to accommodate the characteristics below [3, 4]:

– Intelligence: manufacturing objects consisting of man, machine, and material (3M) collect their data in real time and act autonomously.
– Connectedness: manufacturing objects set up and use connections to other objects of the system for collaboration and for the knowledge and services available on the Internet.
– Responsiveness: manufacturing objects proactively or responsively cope with internal and external changes.

The concepts of these characteristics have been introduced already by much research, such as [5–7]. However, their implementations in reality were still far from reaching

✉ Seung-Jun Shin
sjshin@pknu.ac.kr

1 Information Technology Laboratory, National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA

2 Graduate School of Management of Technology, Pukyong National University, Engineering Building No. 3, 365 Shinseon-ro, Nam-gu, Busan 48547, Republic of Korea

Published online: 20 July 2018

satisfactory outcomes due to practical problems, including the disconnection of physical 3M with the virtual 3M in cyberspace, the difficulty in creating accurate decision-making models (hereafter, "models") customized for individual objects, and the limitation in processing large-sized and various formats of data acquired from those objects [2, 4, 8]. Recently, the convergence of manufacturing technology (MT) with information and communication technology (ICT), such as radio frequency identification (RFID), Internet of Things (IoT), and cyber-physical systems (CPS) is making those characteristics implementable and feasible because current ICT and MT have been advanced considerably to overcome those practical problems [8, 9].

Particularly, the application of CPS in manufacturing, cyber-physical production systems (CPPS), has recently received attention as an advanced SM technology [4]. As CPS are "physical and engineered systems whose operations are monitored, coordinated, controlled, and integrated by a computing and communication core [10]," CPPS aim at implementing autonomous and collaborative manufacturing objects and subsystems based on the context within and across all levels of production [4]. For the pursuit of the three characteristics above, CPPS are required to obtain more-sophisticated capabilities, including improving decision-making and self-optimizing beyond creating transparency and increasing understanding [4]. However, obtaining these capabilities is quite challenging from an implementation viewpoint because CPPS should construct and integrate high-level functionalities, from data acquisition through data analytics to value creation (e.g., prediction, optimization, cognition, and resilient control) [9]. Thus, what to implement effectively and how to implement efficiently these high-level functionalities are growing as critical issues for CPPS.

The platform technology is a key solution to these questions because it provides an integrated cyber-infrastructure where generic functionalities are shareable and any piece of 3M or applications is workable. Such advanced manufacturing platforms can help implement an autonomous and cooperative decision-making environment based on 3M's data, and interconnect the data and models with manufacturing applications including manufacturing execution system (MES) and product lifecycle management (PLM). However, available platforms are lacking (this will be discussed in Section 2.1). The development of an advanced CPPS platform is the goal of the present work.

In the implementation view of such CPPS platforms, three technical challenges appear: *object virtualization*, *data control*, and *model control*. *Object virtualization* refers to identifying physical 3M and translating them into cyberspace to make simulation capabilities available and further improve agility and flexibility [8]. *Object virtualization* is a commonly used concept in CPPS because the virtualized objects, so-called "digital twins," carry out fundamental interactions with physical objects to explore and implement autonomous, cooperative, optimal, and responsive tasks [4]. Without *object virtualization*, physical objects hardly do these intelligent tasks because they are designed to work on their original missions as the top priority. They also have limited hardware and software capabilities in processing such intelligent tasks. Implementing virtualized objects that can alleviate heavy loads on physical objects and further make autonomous and cooperative decisions is essential.

Data control corresponds to data management regarding how data are generated, acquired, stored, and retrieved for supporting intelligent decision-making on 3M. Much research has been done in terms of data models, integrations, and protocols in manufacturing, and the Standard for the Exchange of Product model data (STEP) is a representative example. However, *data control* is an ICT problem and not covered much in the manufacturing realm. *Data control* becomes more significant due to increasing demand for big data, named for the data containing three dimensions: volume (large amounts of data), variety (different formats of data), and velocity (continuous generation of data with high-speed processing) [8]. Machines on shop floors obviously generate a huge amount of data for representing machines' motions, actions, and their operational results (volume). Manufacturing data contain various data models and formats for their own purposes (variety). Manufacturing data that are continuously generated need to be analyzed and used for making decisions in a (near) real-time manner (velocity). Data control is, therefore, an important challenge for making high performance when dealing with manufacturing data.

Model control, which applies the concept of *data control* to models, is used with model management regarding how models are created, stored, retrieved, and even perish. Data itself have no value unless data analysis is performed [8]. Models that use data allow 3M to determine their specific planning accurately and control themselves autonomously. However, models should change along with continuous change in data. In other words, models need to be managed over their lifecycle due to their dependency on data that are sensitive and changeable. For example, when calibration occurs in a measurement device attached to a machine, the previous model would no longer be applicable, and a new model is necessary because the calibration may produce different data values than the old ones. Thus, it is vital to provide the capability for efficient *model control* so that 3M create accurate models in a timely manner and manage models during their lifecycle. Data analytics (DA), which is a new method of data analysis in that it analyzes and mines big data to produce data-driven operational knowledge [11], will be an enabler for *model control*. DA facilitates creating descriptive, prognostic, predictive, and prescriptive models through exploratory solution searching [12]. Particularly, it is known that the implementation of DA as part of CPPS enables machines to continuously track and predict their performance [13].

In view of the above, the present work is motivated by a strong need to develop big data analytics (BDA) platforms in manufacturing. Unlike traditional platforms reported in the literature, the proposed platform incorporates the capabilities of (1) virtualization of 3M for their autonomy and cooperation (*object virtualization*), (2) processing of real and various manufacturing data for mediating physical and virtual 3M (*data control*), and (3) data-driven decision-making for predictive planning on 3M (*model control*). It also enables 3M to perform seamless data exchange using open and standardized data interfaces. In the SM view, it is designed to integrate CPPS with BDA so that it can implement the high-level functionalities encompassing data acquisition, data analytics, and predictive planning as value creation. Hence, the present work is an attempt of designing a CPPS referential architecture that helps achieve intelligence, connectedness, and responsiveness. For such purposes, architecture design for identifying the platform, technical method design for implementing operational mechanisms in the platform, and a prototype implementation for demonstrating the feasibility and effectiveness of the platform is discussed.

This paper is organized as follows. Section 2 describes the design requirements, including a review of the literature and a requirements analysis. Section 3 and Section 4, respectively, present the design of the platform architecture and the technical methods. Section 5 gives an examination of a prototype system, and Section 6 includes a summary and conclusions. To explain the platform and its prototype, a metal cutting process is chosen as the main domain, and energy consumption, a major environmentally conscious metric, is used as the key performance measure.

2 Design requirements

The proposed platform does not simply aim at utilizing BDA techniques in the manufacturing realm, but implementing efficiently the three characteristics mentioned in Section 1. Thus, we should consider capturing requirements in designing the platform architecture. In addition, the three implementation challenges in Section 1 need to be converted to explicit requirements. Section 2.1 includes a discussion of the relevant literature, and Section 2.2 provides the requirements analysis.

2.1 Literature review

The term "platform" in the ICT domain normally means the generic layers, including computers, networks, operating systems, database management systems, user interfaces, system services, and middleware, needed to construct software systems [14]. Meanwhile, in the manufacturing domain, the platform typically represents a software-based framework that supports cooperative works of team members who are separated in time and distributed in space to achieve remote collaboration, distributed control, and data integration across computer-aided technology (CAx) chain [15]. Valilai and Houshmand [15], and Valilai and Houshmand [16] discussed the previous studies relevant to manufacturing platforms, and they classified these platforms into the following groups: (1) *remote collaboration*, (2) *distributed control*, (3) *data integration without STEP*, and (4) *data integration with STEP*.

Remote collaboration provides the technical base for realizing distributed collaborative product development where engineers concurrently and collaboratively participate in design, planning, and manufacturing [17]. Wang and Zhang used a feature-based product model to develop a computer-aided design (CAD) and computer-aided manufacturing (CAM) integrated system to support collaborative work of distributed groups [17]. Nylund and Andresson proposed a simulation-based approach to make autonomous and cooperative entities capable of fulfilling their individual activities in a dynamic environment [18]. Wang presented a Web-based platform for enabling distributed process planning, real-time monitoring, and remote machining [19].

Distributed control exhibits intelligence, robustness, and adaptation through decentralized control to cope with dynamic change and disturbance in manufacturing [20]. An agent-based system, "a computational system that is situated in a dynamic environment and is capable of exhibiting autonomous and intelligent behaviors [21]," has been applied widely. In such a system, agents can encapsulate manufacturing activities or wrap legacy software systems in an open and distributed environment, and represent physical 3M and their negotiation for facilitating distributed control and collaboration. Colombo et al. developed an agent-based control platform for autonomous and cooperative manufacturing systems [22]. Oztemel and Tekez introduced a reference platform to make agents responsible for different manufacturing functions in a distributed environment [23]. Yin et al. presented an agent-based platform to support remote collaboration over the Internet [24]. Lin and Long presented a simulation platform to provide multi-agent-based collaborative and coordinated control [25].

Data integration without STEP deals with product data integration using product information models or data conversion mappers, but it does not follow STEP [15]. Mikos et al. developed a distributed knowledge sharing and reusing platform for potential failure modes and effects analysis using the ontology concept [26]. Meanwhile, *data integration with STEP* focuses on product data integration based on STEP [15]. STEP-based platforms are developed to help ensure data integration and interoperability across various CAx applications [16, 27–30]. As the adoption of cloud computing in manufacturing, called cloud manufacturing, becomes more common, the platform technology provides a fundamental infrastructure to advance the capabilities of searching, mapping, recommending, and executing manufacturing services [31]. Relevant studies, including [32–35], provide demand-driven services with outsourcing ICT resources.

These previous manufacturing platforms have helped provide good solutions for remote collaboration, distributed control, and data integration; however, they are limited in incorporating *object virtualization*, *data control*, and *model control* challenges into their design specifications. Especially, those platforms do not much deal with the interconnection between virtualized objects, their data and models. Their systematic integration can indeed make physical 3M autonomous and cooperative through the utilization of 3M's data-driven customized models adjusted to their real contexts. For clarifying the scope of the present work, Table 1 summarizes the scopes of the previous and present platforms. The first four criteria are taken from [15], and the *object virtualization*, *data control*, and *model control* criteria are added.

2.2 Requirements analysis

The following subsections present the analysis of essential requirements for *object virtualization*, *data control*, and *model control*. We address that the requirements should be met to integrate BDA and CPPS.

2.2.1 Object virtualization

The conventional hierarchical control architecture hardly meets the characteristics in Section 1 because of its inflexibility to respond to changing products and production methods [36]. It also limits the expandability and reconfigurability of manufacturing systems [36]. These limitations inevitably require the control architecture to be autonomous, collaborative, and flexible, which match with the pursuit of holonic manufacturing system (HMS). HMS is a distributed control paradigm using autonomous and cooperating agents, called "holons" [5]. The HMS is regaining popularity as a promising control paradigm in SM, because it originated from realizing the concepts of autonomy (the capability of an entity to create and control its own plans, strategies, and executions) and cooperation (a process whereby a set of entities develops and executes mutually acceptable plans) [4, 5]. In HMS, resource, product, order, and staff holons are autonomous and cooperative building blocks for transforming, transporting, storing, and validating physical objects into a virtual world [5]. To implement HMS, agent technology has been widely used, because agents make it possible to implement efficiently

Table 1 Scopes of manufacturing platforms

Reference	Remote collaboration	Distributed control	Data integration without STEP	Data integration with STEP	Object virtualization	Data control	Model control
[17]	O		O				
[18]	O						
[19]	O	O				O	
[22]		O			O		
[23]		O			O		
[24]		O			O		
[25]	O	O	O		O		
[26]	O		O			O	
[27]		O		O	O		
[28]	O			O			O
[29]				O			O
[16]	O			O			
[30]	O			O			
[15]	O			O		O	
[32]	O			O		O	
[33]	O		O				O
[34]	O		O				
[35]	O		O			O	
Present work	O	O		O	O	O	O

holons that contain a physical part as well as a software part [21, 36].

Implementing agent-based systems in manufacturing systems, in reality, was a big challenge due to the disconnection between physical and virtual objects, i.e., agents. Recently, ICT, such as RFID, IoT, and CPS are making it possible to interconnect these heterogeneous objects easily and efficiently [4]. Meanwhile, in the MT domain, standardized data interfaces such as MTConnect [37] open up the possibility of openness and interoperability of manufacturing data generated on machines [31]. Thus, the HMS concept using agent-based systems is getting closer to being implemented, and the major requirements about HMS are as follows:

[Requirement No. 1] Virtualization: transport physical 3M's objects to virtual objects in cyberspace using agent technology.

[Requirement No. 2] Autonomy and cooperation: enable virtual 3M objects to perform their autonomous and cooperative decision-making across shop floors

[Requirement No. 3] Interconnection: interconnect a series of processes (sensing, acquisition, decision, and control) in the closed control loop of a virtual space with physical 3M objects in real or on time

2.2.2 Data control

Developing an efficient data processing method is essential because data are a mediator of interconnecting physical and virtual 3M objects. In addition, the data are the foundation for the models that must use the data collected from physical objects. When queries for creating models are invoked, a certain data processing method should find relevant data sources quickly, and it should return datasets as the input of models without a time delay. Conventional databases, such as relational database (RDB), hardly meet acceptable performance for big data capability, whereas distributed database (DDB) have better performance because of their parallel-processing ability [38]. The DDB will be key for scale-out (numbers of nodes working together and providing an aggregated performance) because it can cope efficiently with voluminous and various manufacturing data with high-speed streaming. Furthermore, open-source solutions, including the Hadoop Distributed File System (HDFS) [39], MongoDB [40], and HBase [41], are contributing to constructing an easy and cost-effective implementation environment.

Beyond the scale-out capability that commercial vendors mostly address in the above examples, data transparency is critical in the manufacturing domain. It is necessary to access and work easily with manufacturing data, no matter where they are located or who created them. That is, securing data assurance (determining whether the data are correct and come from reliable sources) is important, because the model's correctness largely depends on good datasets being acquired. Data transparency, therefore, is required to provide accessible, readable, and reliable manufacturing data for data assurance. Second, data non-redundancy and consistency are critical factors in data management. Due to the nature of manufacturing industries, many data are scattered across multiple locations, and the data may contain information that are incompatible with specific data or may have to be discarded after the expiration date. Well-thought-out data systems often become too inefficient and complicated to use unless they are regularly managed. Last, data standardization is required for interoperable data exchange. As described in Section 2.2.1, open and standardized data interfaces become good solutions to overcome difficulty in data collection from real shop floors. Moreover, data standardization is important to ensure data quality. Lacking data standardization may cause so-called "bad data" that have negative effects on the model's correctness and reliability by producing unreadable data in some black-boxed machines. In addition, this standardization needs to be applied not only to data but also to models for model sharing and exchange. Thus, the design requirements are the following:

[Requirement No. 4] Scale-out: enable efficiently coping with voluminous and various manufacturing data with high-speed streaming

[Requirement No. 5] Transparency: provide accessible, readable, and reliable manufacturing data

[Requirement No. 6] Non-redundancy and consistency: keep data systems healthy and efficiently

[Requirement No. 7] Standardization: use and analyze high-qualified data and models in an interoperable exchange environment

2.2.3 Model control

In the metal cutting industry, much research has been carried out to develop models for productivity and sustainability performance [42]. Most studies strove to model causal relationships between process planning and Key Performance Index (KPI) due to the great influence of process planning on the efficiency of manufacturing operations. Mostly, the modeling depends on theoretical and/or experimental approaches [43].

The theoretical approach typically uses metal cutting statics with some simplifications. However, it is unable to represent other factors that the metal cutting statics do not reflect, and this results in inaccurate output. For example, Kara and Li pointed out the limitation of the theoretical approach by reporting that theoretical energy consumption models do not clearly quantify specific process energy, and thus, their energy prediction is infeasible unless they provide the exact value for each variable and coefficient [43]. Meanwhile, the experimental approach conducts experiment within design of experiments (DOE), "the application of geometric principles to statistical sampling to obtain desired results together with minimizing the number of experiments [44]." This empirical approach can provide more reality and practicability than the theoretical approach; however, it derives statistical models that are only effective within the boundary covered by a small number of experiments. It also hardly covers the diversity of manufacturing conditions in real shop floors, where many manufacturing conditions exist and dynamically change.

One solution to overcome the limitations of the traditional approaches is to use real and historical data that have been collected and accumulated from previous manufacturing operations. We call this approach "data analytics modeling approach," because it is associated with the pursuit of DA (see Section 1), which depends on real data and is independent of theoretical and experimental data (these are still important when identifying the data collected and designing model structures). This approach, as an empirical approach, creates models by identifying the data needed, collecting relevant historical datasets (this step largely relates to *data control*), and generating and validating empirical models from the datasets. It calculates accurate outputs because it provides the exact value for each variable and coefficient by using real data rather than assumptions. It can also create multiple and machine-specific models that cover the diversity of manufacturing conditions, because it depends on the data collected from previous operations, independently of DOE.

The DA approach can be more beneficial when it achieves model granularity, which scales down the decomposition of models at the designated minimum level. The model granularity leads to precise and flexible modeling because it can decompose and re-compose models dynamically in terms of stratification. For example, an energy prediction model dedicated to a machining feature can anticipate an energy value for the feature; however, it cannot work when the feature geometry changes. If energy models are created in terms of tool paths (the minimum criterion is the type of tool path), the changed energy value can be obtained through the appropriate re-composition of these models aligning with tool paths on the changed feature. Hence, the design requirements are:

[Requirement No. 8] Data-driven model: apply a DA modeling approach, creating models by identifying the data needed, collecting datasets, and generating and validating models.

[Requirement No. 9] Model specificity: create models specific to and customized for dedicated machines based on their own data

[Requirement No. 10] Model granularity: scale down the granularity of models for precise and flexible modeling

3 Architecture design

Based on the design requirements in Section 2, we design a BDA platform. In Section 3.1, we introduce its conceptual architecture, and in Section 3.2, we present its detailed functional architecture.

3.1 Conceptual architecture

Figure 1 shows the conceptual architecture. The proposed platform is designed to incorporate intelligence, connectedness, and responsiveness described in Section 1. The fundamental idea is that the virtual 3M mirrored with their physical objects gain data insights through the interconnection of their data and models. These data and models have been generally dealt with in the DA domain, whereas those virtual objects in the industrial control domain. By coupling virtual objects and their data and models tightly, the proposed platform pursues the implementation of an autonomous and collaborative CPPS over the traditional industrial control, which limits executing restricted instructions automatically. Based on the requirements addressed in Section 2, this architecture consists of three parts (virtual shop floor, data warehouse, and data analytics center) for carrying out the mechanisms below.

The main mechanism of the virtual shop floor is (1) to perform the necessary decision-making using status data and models, at present, (2) to plan or control a physical shop floor by the input of decisions made, and (3) to monitor model performance and report it to the data analytics. The main mechanism of the data warehouse is (1) to preprocess sensor and status data coming from a physical shop floor, (2) to store the shop floor data refined, and (3) to provide the data analytics center with datasets for model creation. The following is the main mechanism of the data analytics center: (1) to create models and optimal strategies for individual agents to meet the goals given by a parent application and (2) to manage the model lifecycle based on the performance report. Besides the fundamental mechanisms above, the assistant mechanisms include (1) managing agents during their lifecycle, (2) governing the quality, uncertainty, and lifecycle of manufacturing data, (3) processing automated workflow for timely creation and use of models, and (4) controlling security for protecting proprietary models and data.

3.2 Detailed architecture

Figure 2 presents the detailed architecture based on the conceptual architecture illustrated in Fig. 1. The following items explain rationales and functions of the designed modules. In Section 4, we will describe the technical methods needed to implement the virtual shop floor, data warehouse, and data analytics center as the core modules of the platform.

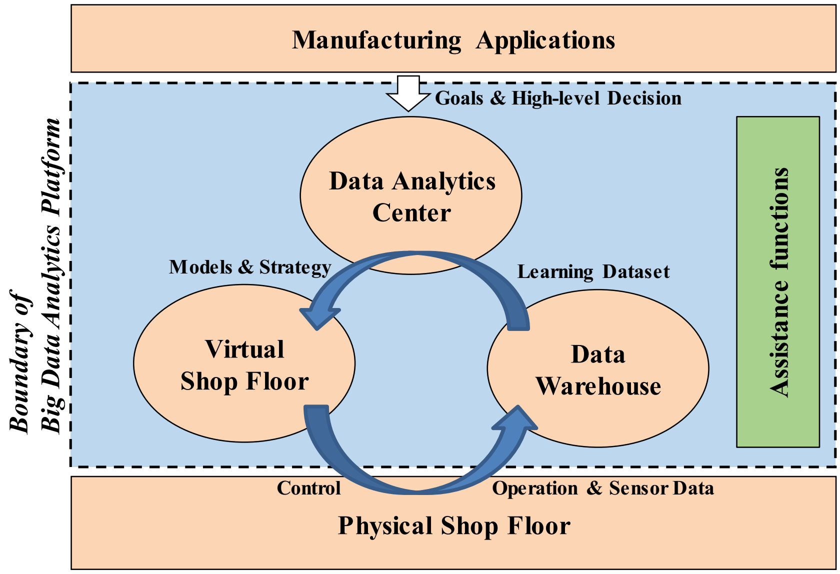


Fig. 1 Conceptual architecture

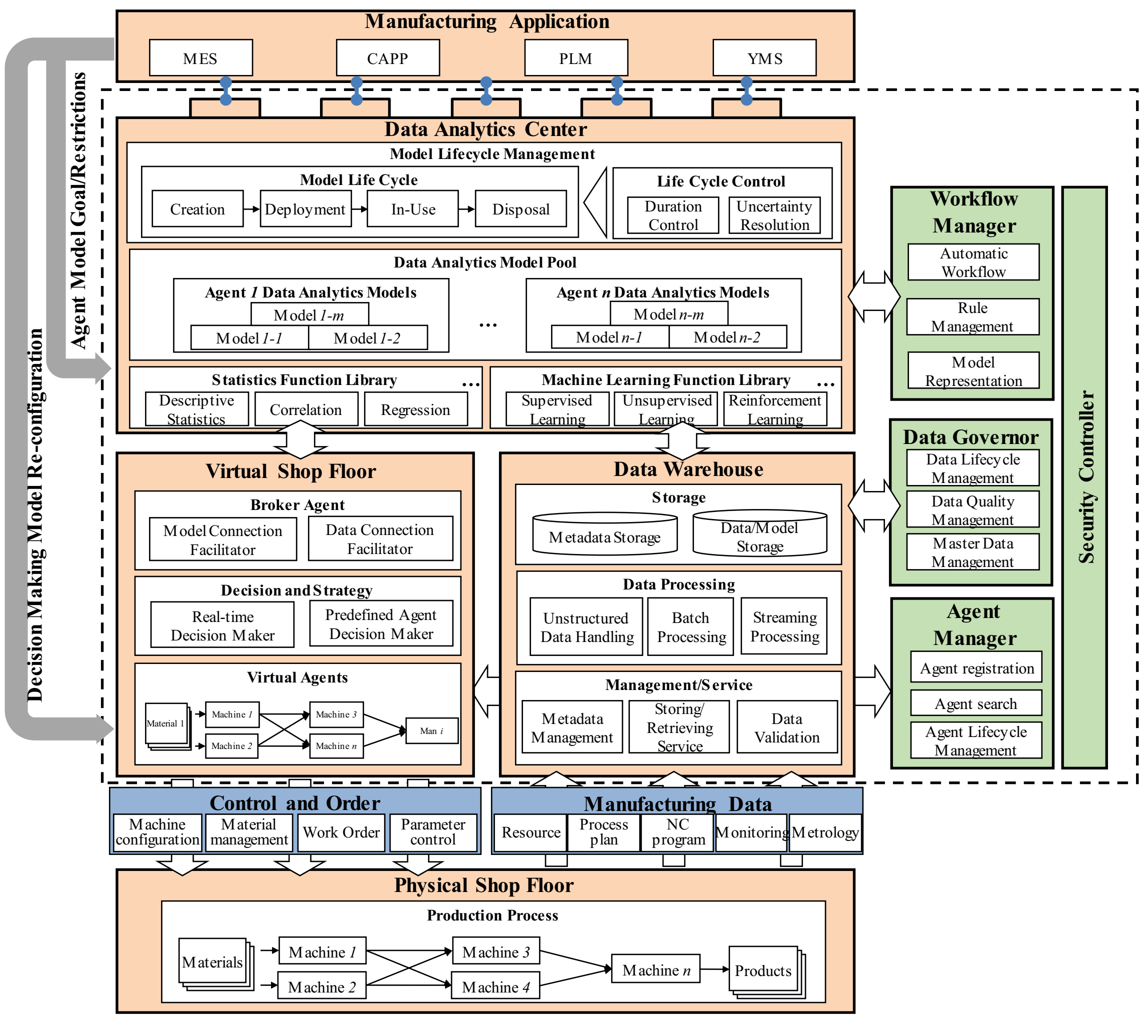


Fig. 2 Detailed architecture

- *Physical shop floor*: This is the shop floor where real production takes place as the creating point of manufacturing data. Physical 3M objects perform the work assigned for achieving goals under given process planning and configurations. They simultaneously output many operation and sensor data that involve resources, planning, control, monitoring, and metrology. Data collection systems, like supervisory control and data acquisition systems, gather and transmit such collected data into a data warehouse. Open data interfaces, such as an MTConnect agent, i.e., software that receives and stores a time series of data samples or events and acts as a bridge between a machine and a client application [37], support efficient data collection.
- *Virtual shop floor*: This is the virtualized shop floor of a physical one using agent technology. Each agent has information regarding its IDentification (ID), authorization, configuration, capability, and mission. It also identifies the operation and status data needed to collect from its corresponding physical object, and transmits them and their metadata, i.e., the data representing a source, time, means, purpose, type, and size of raw data, to the data warehouse. In addition, it autonomously decides individual planning and control by accessing the model through a broker agent, i.e., a model connection facilitator. Multi-agents, a set of agents, communicate together in a hierarchical and/or heterarchical manner and make decisions cooperatively to achieve given goals. (relevant requirements: no. 1, no. 2, no. 3, no. 5, no. 7, no. 8, no. 9)
- *Data warehouse*: This is an information hub that stores and exchanges manufacturing data. This warehouse receives raw manufacturing data and their metadata from the physical shop floor and stores them structurally in terms of designated attributes. It then processes the raw data and returns datasets when queries are called by the data analytics center. In addition, it transmits the current state of the physical shop floor to the virtual shop floor in real time, enabling synchronization between both shop floors. Basically, standardized manufacturing data are exchanged to ensure data interoperability. (requirements: no. 4, no. 5, no. 6, no. 7, no. 8)
- *Data analytics center*: This manages the model's creation, storage, retrieval and uncertainty over the life cycle. Regarding the duration of the model, this center continues to observe the accuracy of the model and determines whether the model expires when it detects a drop in accuracy below a certain threshold. It retrieves a workflow to create a model that can substitute for the expired model. The uncertainty is used to measure the difference between a model and its respective system [45]. Uncertainty arises from natural variability and information uncertainty due to poor data and modeling uncertainty induced by assumptions and approximations [46]. Because determining the accuracy and duration of the model is important, the uncertainty of the model needs to be quantified. The center also provides machine-learning, statistical, or stochastic-based models that build on mathematical functions needed to create data-driven models. Each agent retrieves such models through a broker agent and decides predictive operations and controls, based on the results that models output. Here, the DA approach is applied to specify a structural and logical procedure for model creation. (requirements: no. 2, no. 5, no. 7, no. 8, no. 9, no. 10)
- *Manufacturing application*: This represents existing manufacturing applications, such as MES, PLM, and computer-aided process planning (CAPP) systems. These applications need to communicate with the platform through their application interfaces because they eventually supervise and manage all activities and events occurring on the physical shop floor. Decisions made by manufacturing applications are then delivered to the physical/virtual shop floors in real time or on time. For example, a manufacturer that uses a CAPP system can build predictive planning by determining controllable parameters (e.g., feedrate, spindle speed, and cutting depth) whose energy can be foreseen by relevant models. (requirements: no. 2, no. 3, no. 5, no. 8, no. 9)
- *Agent manager*: This registers agents with their ID, searches adequate agents, and manages them during their lifecycles. Each agent must register with its ID in an agent directory, so that agents are retrieved from the directory for achieving specific goals. Some agents may perish in the directory due to expiration (e.g., when an old machine agent is replaced by a new one). Agents should be managed throughout their lifecycle as with model lifecycle management. (requirements: no. 1, no. 2, no. 3)
- *Data governor*: This manages master data as well as the lifecycle and quality of raw data. Data lifecycle management is necessary in a high-volume data environment, because indiscriminate and permanent data retention inevitably increases data-archiving cost in database systems. Because data quality directly relates to the model's performance, as addressed in Section 2.2, data refinement by data cleaning rules or algorithms needs to be applied to provide high quality of data. (requirements: no. 4, no. 5, no. 6, no. 7)
- *Workflow manager*: This controls workflows to automate the tasks performed on the platform, manages the rules designed to handle workflow appropriately, and engages in model representation. The manual effort to create, use, and terminate models is formidable, which makes the platform inefficient. Workflow automation helps achieve optimal operations in the platform, providing the capabilities of connecting data and models, integrating the platform with manufacturing applications, assigning software resources, and progressing unit-tasks under given rules.

The rules ensure the task sequence in such a way that the next task is invoked or a user is notified once the former task is complete. (requirements: no. 6, no. 8, no. 9, no. 10)

- *Security controller*: This protects against computer viruses and hacking, and controls electronic authorization and authentication. Data and models that incorporate manufacturing experience and knowledge are valuable and, thus, must be protected. Security control is important, even in the environment where open and interoperable data exchange takes place on the platform. Such firewalls are required to ensure the protection of data and models. Authenticated users must access and maintain data and models through rigorous authentication procedures. (requirements: no. 3, no. 5, no. 6, no. 7)

4 Implementation method

This section describes the implemental aspect, i.e., how to implement the platform technically. This section introduces technical methods for the three core modules: virtual shop floor, data warehouse, and data analytics center.

4.1 Virtual shop floor

Many studies have highlighted the effectiveness of multi-agent systems (MAS) in HMS by demonstrating the benefits of material handling, planning, scheduling, and control [36, 47]. MAS have also evolved into data-driven intelligence, distributed control, and system integration with virtual factory and manufacturing applications [20]. Considering these features, MAS are effective and efficient to implement in HMS, and they can underlie implementing a virtual shop floor.

Figure 3 presents a use case of the virtual shop floor. It needs to reach dynamic manufacturing intelligence in HMS because the HMS should arrive in real or on time for decision-making and flexibility. This can be achieved by the integration of data-driven models. For example, predictive planning allows 3M objects to decide their process plans based on the anticipated KPI using the prediction models that have been generated from their data. The virtual shop floor also needs to achieve multi-level optimization, i.e., the optimization of multiple KPIs in multi-levels of a manufacturing system. Depending on 3M objects' decisions and negotiations, each object optimizes multiple KPIs for itself at the machine level. A production line consisting of 3M objects successively optimizes multiple KPIs at the line level. An example is that a machine tool explores process parameters for minimizing both machining time and energy, and then a shop floor allocates an optimal path to minimize production time and energy. In another view, the virtual shop floor needs to embody systems integration efficiently by using the advantage of agent technology. *N*-to-*N* implementation as a traditional approach has no linking point; thus, much cost and time are spent as the number of applications (*N*) increases. In the virtual shop floor, MAS using a standard communication protocol can be the linking point for 1-to-*N* implementation across heterogeneous applications.

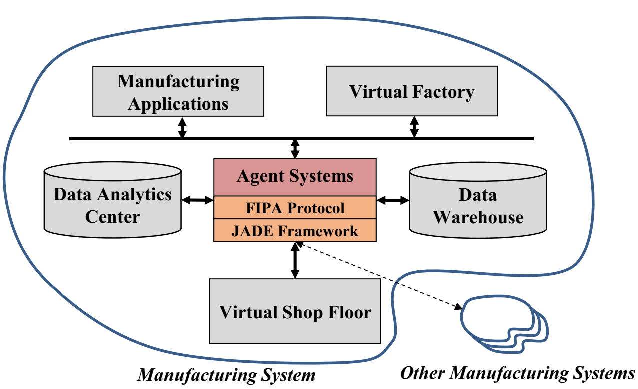


Fig. 3 Use cases of multi-agent systems in virtual shop floor

The Foundation for Intelligent Physical Agents (FIPA) can be used to implement such virtual shop floor. FIPA is the standardized protocol to produce MAS specifications for supporting interoperability, open service interaction, and heterogeneous development [48]. The FIPA-based MAS communicates and operates within cooperation domains, conforming to the concept of HMS [22]. FIPA also allows interconnection between agents and external applications through Web services within the Java Agent Development Framework (JADE), a widespread agent-oriented middleware complying with FIPA [49].

Based on the use cases above, a structure of the virtual shop floor is designed, as shown in Fig. 4. This structure builds on the HMS referential structure, which contains basic holons, i.e., agents, consisting of resource, product, and order [5]. Furthermore, we add two distinct design factors in the system design, compared with conventional HMS structures. First, two staff agents are added to interconnect the basic HMS agents with the data warehouse and data analytics center. In conventional HMS structures, staff holons have been designed to carry out evaluation, mediation, management, and coordination to support the basic holons that intensively work for achieving goals such as task allocation, fault-tolerance, and scheduling [21, 36, 47]. However, there were few staff holons to interconnect with data repositories for extracting data, as well as model repositories for creating and retrieving models. Second, basic holons are redesigned to obtain the capability of making autonomous and collaborative decisions using their dedicated data and models. We intensify performance prediction/optimization and disturbance handling in the basic holons' functions beyond their typical functionalities, including identification, monitoring, management, and execution.

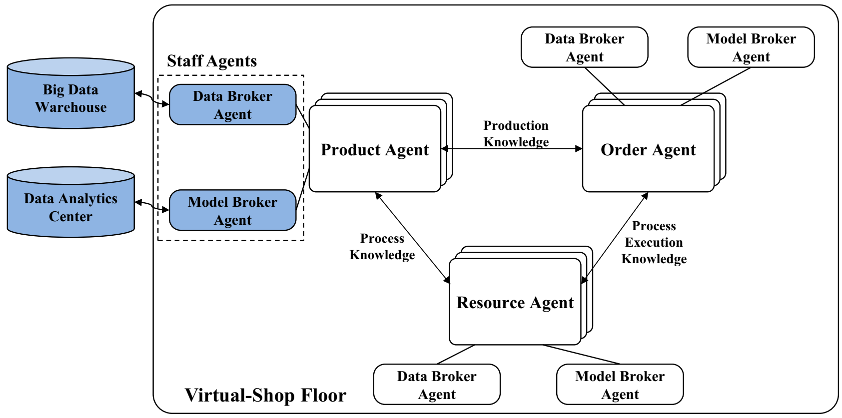


Fig. 4 Structure of virtual shop floor

Figure 5 shows the operational flow that identifies functions and interactions of individual agents inside the virtual shop floor. This flow basically complies with the FIPA Contract Net Interaction Protocol [49]. This protocol builds upon the Contract Net Protocol (CNP), a high-level negotiation protocol for agent-based manufacturing systems [50], to realize autonomous and collaborative decision-making between the three basic agents. We improve the FIPA CNP by adding the functions interactive with the data and model broker agents to accommodate prediction, optimization, and disturbance handling.

In Fig. 5, when an order agent announces and proposes a task requested from a product agent, each resource agent checks its availability. The resource agent, for itself, predicts the KPI and finds an optimal process plan with regard to the announced task, using product and process planning data delivered from a data broker agent as well as prediction and optimization models delivered from a model broker agent. Available resource agents generate counterproposals for grasping the task and feed them forward to the order agent. The order agent evaluates the counterproposals to select the best resource agent and carries out remaining functions with

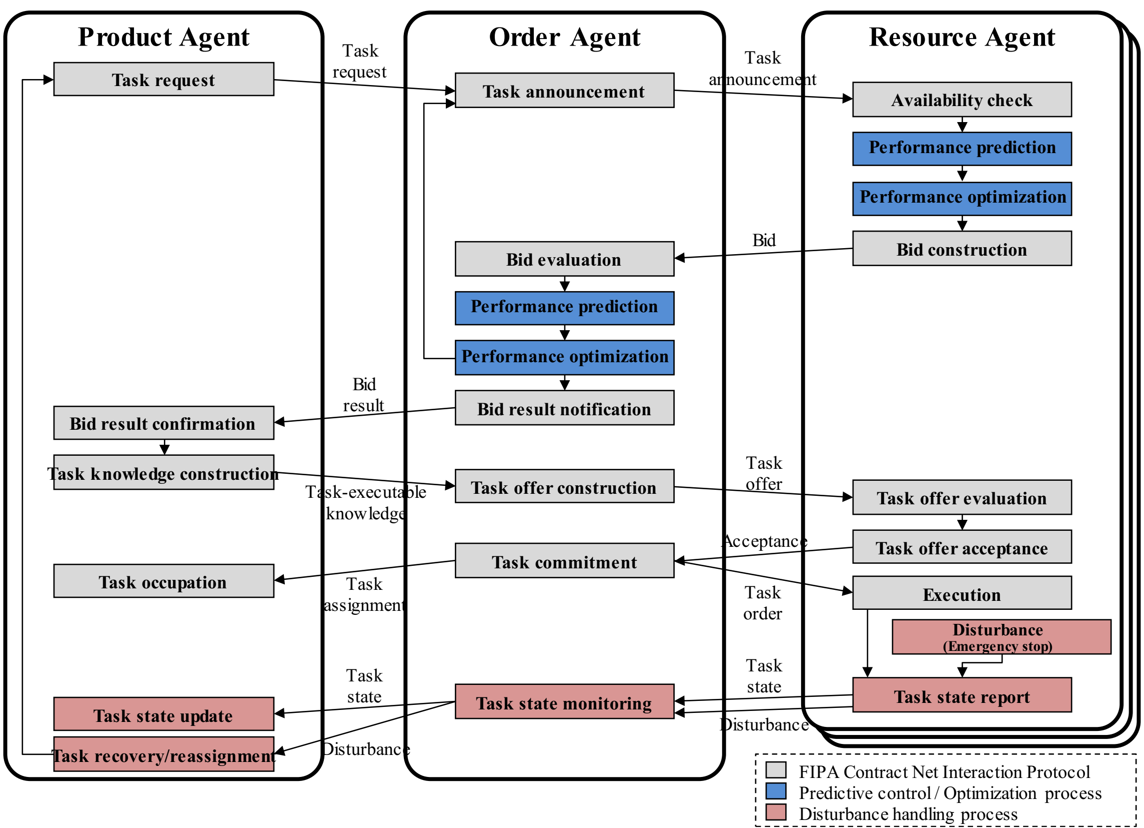


Fig. 5 Operational flow chart for product, order, and resource agents

communication of product and resource agents. Because an order agent gathers the proposals transmitted from available resource agents, it can have visibility for performing higher-level prediction and optimization in a production line. If an emergency occurs in a resource agent who is executing the task, the agent immediately stops and reports the task to order and product agents who are monitoring the task. The product agent updates the task state, and the order agent reassigns the task to another resource agent in the same manner.

Table 2 identifies the functions and relevant data on all these agents to make the association with the virtual shop floor. Here, a "domain" represents the applications of agents or the modules connected via staff agents. "Function" means the functions that the agent should perform. "Relevant data" indicates the data associated to perform their functions with some relevant standards (bracket).

Figure 6 presents the schematic structure of the virtual shop floor for the systems integration by means of 1-to-*N* implementation using JADE. This structure, re-edited from the architecture of the Web Service Integration Gateway (WSIG) [49], identifies the fundamental components and the operational protocol in JADE to communicate with external applications. Because the WSIG offers bidirectional interconnectivity between agents and Web services [49], the three basic agents are allowed to exchange individually their information with external applications, and vice versa. The JADE Directory Facilitator (DF) registers, deregisters, modifies, and searches agents. The JADE Gateway Agent (GA) receives and translates agent service or Web service registrations, respectively, from JADE DF or Universal Description Discovery and Integration (UDDI), which is the directory service supporting the description and discovery of Web services and their providers [51]. JADE GA also receives and processes agents and Web service invocation. The ACL<>SOAP Message Codec translates Agent Communication Language (ACL) messages into Simple Object Access Protocol

Table 2 Domain, functions, and data of agents

Agent type	Resource agent	Product agent	Order agent	Data broker agent	Model broker agent
Domain	Production resource	Part, work-In-process, product	Orders of customer, make-to-Stock, maintain	Data warehouse	Data analytics center
Common function	- Identification (RFID, Barcode, Universal ID, IoT, etc.) - Communication (FIPA, IPv6, TCP/IP, etc.) - Data creation - Data retrieval - Model creation - Model retrieval - State monitoring - Reporting			- Access authorization - Workflow control	
Stand-alone function	- Health diagnosis	- Task request	- Planning	- Data transmission	- Model transmission
	- Predictive control	- Task verification	- Scheduling	- Data collection	- Model collection
	- Optimal control	- Task solving	- Prediction	- Metadata collection	- Model criteria transmission
	- Verification - Resource allocation - Resource execution - Measurement		- Optimization - Process alignment - Disturbance handling - Verification - Measurement	- Metadata supply	- Model criteria collection
Relevant data	- Machine specification (ISO14649, ISO15531) - Tool specification (ISO13399, ISO14649) - Machine-execution (ISO6983) - Process plan (ISO14649, ISO10303–238) - Machine-monitoring (MTConnect, OPC-UA) - Measurement (ISO10303–242)	- Product design (ISO10303–203) - Process plan (ISO14649, ISO10303–238) - Quality (ISO10303–242) - Product lifecycle (ISO10303–239)	- Production planning (ISO10303–240) - Resource (ISO15531, ISO14649) - Order (ISO10303–240)	Data (JSON)	Model (PMML, OPL, MML)

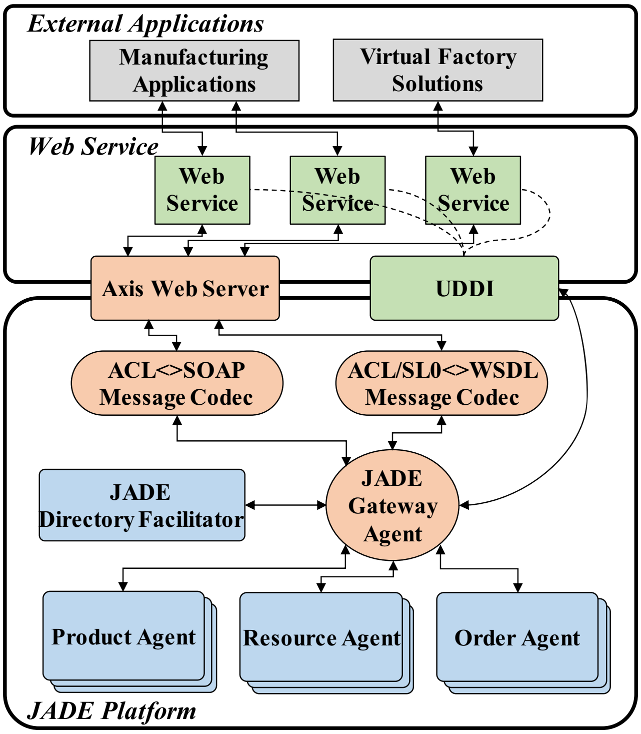


Fig. 6 Systems integration structure on virtual shop floor (re-edited from [49])

(SOAP) messages, and vice versa. The ACL/SL0<>WSDL Message Codec translates FIPA-ACL/SL0 service descriptions into Web Services Description Language (WSDL), and vice versa. The Axis Web Server is the server to send and receive SOAP messages to and from Web services [49].

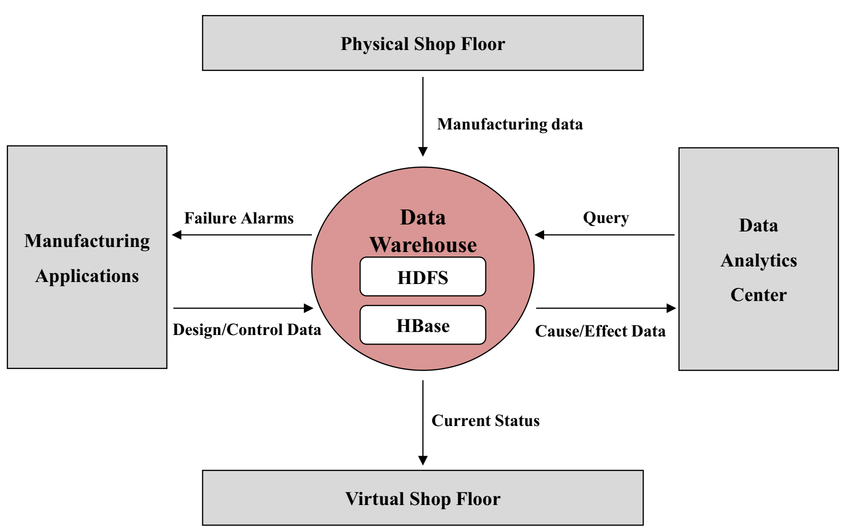
4.2 Data warehouse

Figure 7 shows a use case of the data warehouse. The data warehouse is responsible for storing and managing data produced and used by all the modules. Its main work is to store design and control data from manufacturing applications, to gather and store manufacturing data generated from the physical shop floor, to provide datasets needed to create models in the data analytics center, and to provide current states to the virtual shop floor, as mentioned in Section 3.2.

In this module, it is important to clarify the relationship between design/control data (such as CAD, CAPP, and CAM data) from manufacturing applications and manufacturing data (such as MTConnect data) from the physical shop floor. This module extracts metadata from shop floor data (e.g., MTConnect) that come up in real time, finds out which design/control data correspond to the shop floor data, and stores this relation as pairwise cause–effect datasets. HBase is used to make this metadata analysis possible. As HBase is a highly scalable, open-source, and non-relational database that runs on HDFS [41], it maintains performance while scaling out to hundreds of nodes, supporting billions of rows and millions of columns. Additionally, HBase adds a timestamp to each cell and can keep previous versions, allowing applications to store and access the lineage of a dataset easily. This capability makes it possible to collect easily and quickly up-to-date data into collections based on metadata; hence, this technology is suitable for a manufacturing environment that contains many sensors and generates data at very high speeds on each sensor at work.

Figure 8 presents a technical framework of the data warehouse. Here, large-sized and various formats of data, even if they are standardized data, inevitably decrease the efficiency of data handling, because they are originally formalized for representing their designated contents rather than supporting efficient data handling. Thus, there is a need to use a unique and simple way for unifying data representations. For this purpose, Java Script Object Notation (JSON), a lightweight data-interchange format built on a collection of name–value pairs and an ordered list of values [52], is used. For example, a STEP-compliant data

Fig. 7 Use cases of data warehouse



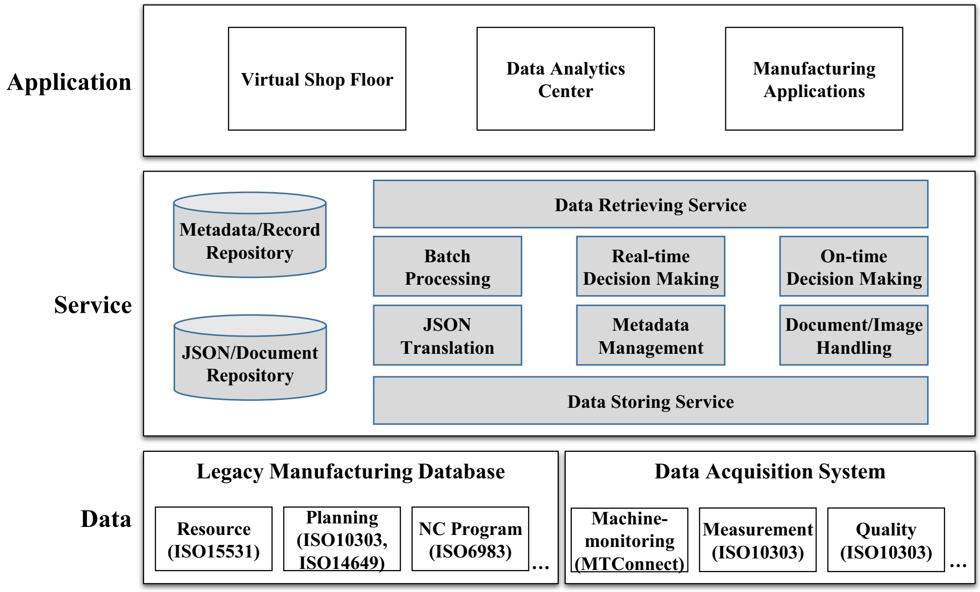


Fig. 8 Technical framework of data warehouse

interface for numerical controls (STEP-NC) part program, which identifies process planning for machining operations, is transformed to the JSON format for making it readable and usable. Therefore, the original raw data and their associated JSON-formatted data are stored in the data warehouse. Table 3 explains the details of the components included in the service layer in Fig. 8.

4.3 Data analytics center

The objective for the data analytics center is to create and manage data-driven models. Thus, this center needs to incorporate a logical modeling approach for creating models and provide a systematic structure for managing models. The following subsections, respectively, describe the modeling approach and the structure design.

4.3.1 Data analytics modeling approach

As mentioned in Section 2.2.3, the data analytics modeling approach aims at creating machine-specific and granular models using manufacturing data. Some good approaches, such as CRISP-DM, which defines a standardized process model for data mining from a business perspective [53], have been introduced; however, they are not specific to the manufacturing domain. The proposed approach identifies a logical modeling procedure for creating models based on machine-learning, statistical, or stochastic analysis, as shown in Fig. 9. Here, "component model" means the model that figures out a numerical relationship between cause-and-effect (CE) data up to the designated level and, thus, can predict target performance at a certain manufacturing configuration.

1. Process data attribute (PDA) identification: This identifies the data attributes needed to collect manufacturing data. These data attributes comprise manufacturing configuration (MC) and CE data attributes. The MC attributes specify a process context where a component model

Table 3 Components and functions in data warehouse

Component	Function
Data storing service	Provide the data interface to collect raw manufacturing data from legacy manufacturing databases and data acquisition systems
Data retrieving service	Provide the data interface to invoke queries and return datasets with applications
JSON translation	Formalize raw manufacturing data into JSON format, and transmit the JSON-formatted data to JSON/Document repository
Metadata management	Generate and manage metadata to identify manufacturing data for data querying and retrieving
Document/image handling	Handle unstructured data formats such as image and PDF files
Batch processing	Process large datasets to respond to the queries that request a series of jobs at one time
Real-time decision-making	Support real-time control by monitoring abnormal data signals
On time decision-making	Support on time control derived from applications
JSON/document repository	Store raw manufacturing data and their JSON-formatted data
Metadata/Record repository	Store the metadata that associate with retrieving raw and JSON-formatted data from JSON/Document repository

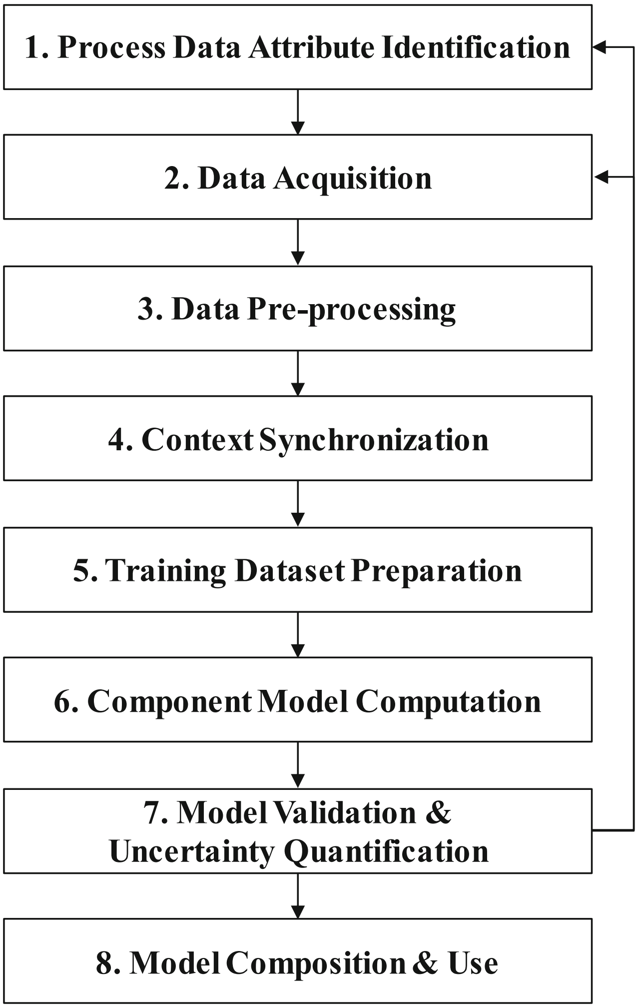


Fig. 9 Procedure of data analytics modeling approach

can be applied. That means the identical MC set uses the same component model; meanwhile, a different MC set requires another model. A machine, cutting tool, workpiece material, coolant, and machining operation can be MC attributes, because a selection of those attributes determines a specific machining context. For example, when the attribute of workpiece material changes in an MC set, a model for the old material is not valid and should be replaced by a new one for the new material, because the latter material has different material properties from the former one. The appropriate identification of MC attributes will decide the granularity of models, as explained in Section 2.2.3. Meanwhile, the CE attribute identifies the data attributes collected to make numerical CE relationships under a set of MC attributes. The CE attributes correspond to input-and-output variables, and their instantiated datasets make it possible to generate numerical functions, $y = f(X)$. Due to influences of process plan decisions on machining performance, as known from the literature, process parameters—feedrate, spindle speed, and cutting depth and width—can be cause attributes (X variables); on the other hand, KPIs such as energy, machining time, and surface roughness can be effect attributes (y variable). For example, given a set of MC attributes, Eq. (1) expresses an energy component model where f and ε (error term) are derived from the numerical relationships mined from instances of CE attributes.

$$\text{Energy} = f(\text{feedrate, spindle speed, cutting depth, cutting width}) + \varepsilon \quad (1)$$

2. Data acquisition: This searches and collects the instances of MC and CE attributes from raw manufacturing data stored in the data warehouse. Creating models invokes a query for requesting necessary data instances from the data warehouse that searches the data using their metadata and returns them in accordance with a requested data format (the details are in Section 4.2).
3. Data pre-processing: This is a process of cleaning, normalizing, and transforming data to make datasets of high quality. Preparing training datasets is necessary for machine-learning and statistical analysis so that a computer directly acquires knowledge and learns to solve problems. Machine-learning and statistical analysis relies heavily on training datasets. Because manufacturing data typically contain erroneous or missing data, data cleaning is needed to eliminate or correct these data, thereby decreasing the data uncertainty mentioned in Section 2.2.2. Data normalization scales down the large difference between maximum and minimum values appropriately using min–max or z-score normalization [54]. Data transformation augments the space of data attributes by inferring or creating additional data attributes [54].
4. Context synchronization: This matches and synchronizes MC and CE data attribute instances on timestamps to make training datasets. MC and cause attributes are generally located in process planning and NC programming data formalized into time-independent and structural data types, whereas effect attributes are found in machine-monitoring data, which typically form in a time series type. These two heterogeneous data sources need to be merged into one dataset, because machine-learning and statistical analysis require training datasets consisting of tuples of cause data and their corresponding effect data. In metal cutting, machine-monitoring data (effect data) represent snapshots of the observation and measurement of a machine's actions on timestamps. They need to be synchronized along with their process planning and numerical control (NC) programming data (cause data), because

certain actions of a machine are not understandable without these cause data.

5. Training dataset preparation: This divides the entire training dataset into separate datasets in terms of the combination of MC attributes. As a set of MC attributes determines a component model, MC attributes can be used as the classification criteria to divide into a set of datasets, as shown in Fig. 10. Each training dataset contains multiple CE data tuples given a set of MC attributes.
6. Component model computation: This computes numerical functions by machine-learning or statistical analysis using training datasets, and it results in creating component models. Because a numerical function represents the relationship between CE data attributes, it can be used to predict the KPI by a certain input of process plan decisions, as illustrated in Fig. 10. For example, an energy value can be calculated from the input of process parameters once we obtain a numerical function where the energy value corresponds to y and the process parameters to X. Many machine-learning techniques, such as the support vector machine, Bayesian network, and artificial neural network (ANN), are available (ANN is used in Fig. 10). Equation (2) expresses the mathematical notation associated with the ANN-based model (Fig. 10). Statistical or stochastic analysis can be similarly applicable in computing such numerical functions because both analyses are widely accepted to predict future values based on probability theory. For example, a regression technique, as a statistical analysis, provides explicitness and usefulness for computing statistically significant relationships between input and output parameters [55]. Equation (3) expresses an example of a polynomial regression model, which has the same objective but a different structure from Eq. (2). In several stochastic techniques, for example, the Markov process estimates performance measures through historical data [56].

$$y = f_o\left(\sum_{j=0}^{p} w_{oj} f_h\left(\sum_{k=0}^{q} w_{jk} x_k\right)\right) + \varepsilon \tag{2}$$

where p, q: the number of neurons (i.e., information-processing units) at each layer, w_{oj} and w_{jk}: weight values between layers, f_{o} and f_{h}: activation functions.

$$y = \alpha_0 + \sum_{i=1}^{b} \alpha_i x_i + \sum_{j=1}^{b} \alpha_j x_j^2 + \varepsilon \tag{3}$$

where α: coefficients of terms, b: the number of x variables.

7. Model validation and uncertainty quantification (UQ): This validates the reliability of models and quantifies the uncertainty of the models. The numerical functions computed require measurable validation to ensure that their performance is above a satisfactory threshold. Those functions can be validated through the cross-validation technique that partitions the data into training or validation datasets to check the model's performance in an iterative way. The correctness of the functions can be measured using root-mean-square error (RMSE), which analyzes the differences between predicted and measured values. UQ needs to be performed to characterize the sources of the uncertainty, and measure the probability distribution of models [46]. The uncertainty in such computed models can be generally represented as model form error ($\varepsilon_{\mathrm{model}}$) and numerical error ($\varepsilon_{\mathrm{num}}$) [57]. $\varepsilon_{\mathrm{model}}$ relates to whether models and their parameters correctly reflect the real phenomenon. $\varepsilon_{\mathrm{model}}$ can be evaluated using calibration and validation data, based on the comparison of model prediction against physical observation [46]. Meanwhile, $\varepsilon_{\mathrm{num}}$, i.e., solution approximation error, involves the errors in solving the computational models itself such as discretization or surrogate model error [46]. The discretization error can be quantified by comparing solutions with different levels of discretization

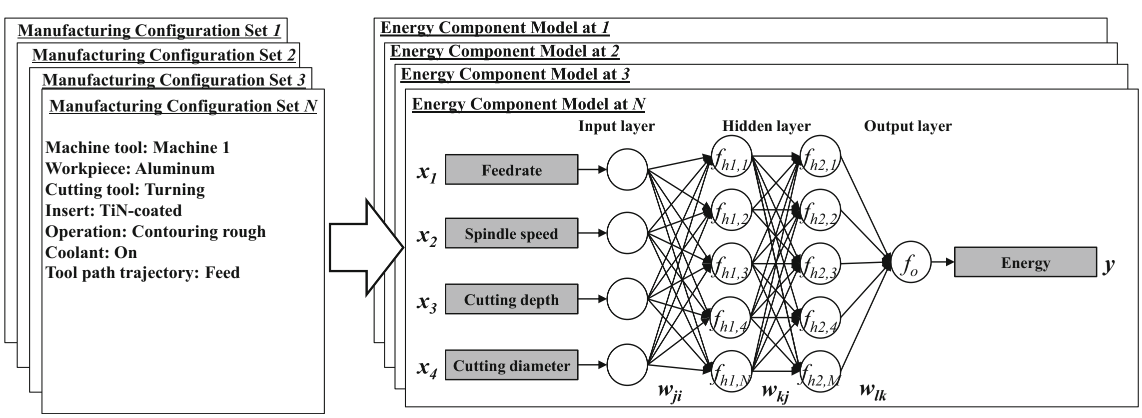


Fig. 10 Relations of manufacturing configuration sets with component models

while the surrogate model error can be estimated by comparing the output of the surrogate model along with multiple running [57]. Once UQ is made, it can be used to decide the lifespan of models. If model uncertainty (error values) exceeds a certain threshold, the model can be invalid and thus a new model should substitute for the old model using different model forms or new training datasets. If $\varepsilon_{\text{model}}$ or ε_{num} exceeds a threshold, the DA procedure can be preferably re-taken from the first step (PDA identification) or the second step (data acquisition), respectively.

8. Model composition and use: This composes and uses models for making decisions in process planning and machining stages once reliable models are obtained. Composing sequentially individual component models in regard to sets of MC attributes can derive the KPI. For example, component models can be composed to predict the energy consumed during the execution of an NC program, because each code block on the NC program associates with a set of MC attributes.

4.3.2 Model structure

Model structuring is significant because the structure decides how efficiently a group of component models is segmented and classified [12]. Efficient design and operation of hundreds or even thousands of models can be promoted through the creation of a well-organized structure, which can configure the model pool in Fig. 2. Figure 11 shows the concept of the model structure. This structure contains a group of individual component models, which are classified by the MC attributes set. The structure constructs the common model pool that combines building blocks (component models). Here, such model standardization can enhance the model's interoperability. Predictive Model Markup Language (PMML), which provides a standardized structure and format to represent regression and machine-learning functions [58], can be used for representing models in a standardized way. By these, the structure will enable common, reusable, and extensible applications of component models across shop floors.

5 Prototype implementation

To demonstrate the present work, this section describes prototype systems that we are implementing. It shows clearly that the data analytics modeling approach is demonstrated in advance of the data warehouse and virtual shop floor. The warehouse plays the role of a data supplier for acting the approach, and the shop floor acts as a model user. In Section 5.1, the data analytics modeling for energy prediction and machining time prediction is explained. Section 5.2 and Section 5.3, respectively, present the prototype of the data warehouse and virtual shop floor.

5.1 Data analytics modeling in machining

This section demonstrates the data analytics modeling approach described in Section 4.3. This approach is applied to two different applications to check the effectiveness of the approach in multiple applications and, thus, to confirm the

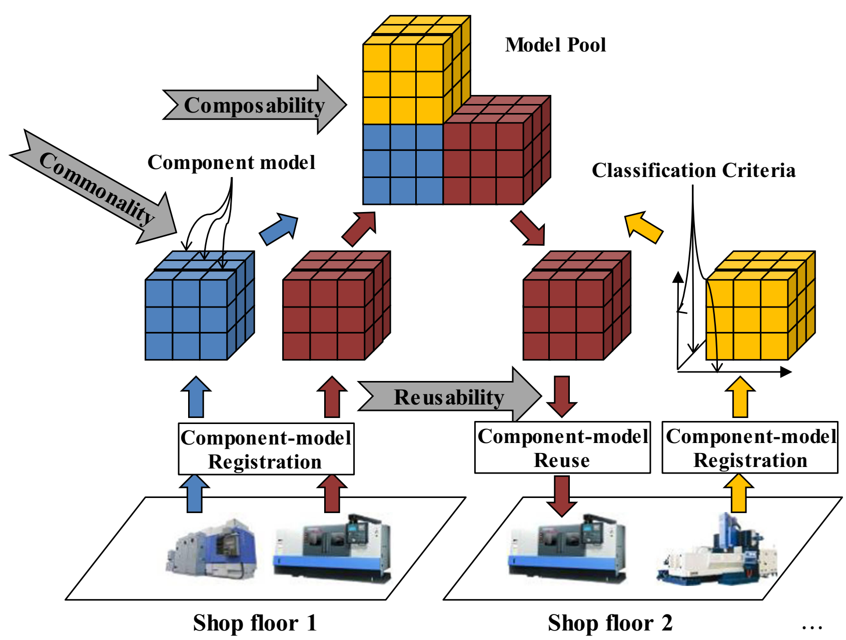


Fig. 11 Concept of model structure

generality of the approach. In Section 5.1.1 and Section 5.1.2, respectively, we describe energy prediction modeling for energy-efficient machining and time prediction modeling for production-efficient machining.

5.1.1 Energy prediction

Actual machining was carried out to create energy prediction models (hereafter, energy models) to anticipate the energy consumed during the execution of an NC program for producing a 2.5-dimensional milling part. Figure 12 shows a machined part and its tool path trajectories. Table 4 lists sets of process parameters for producing 12 parts. Here, three process parameters—feedrate, spindle speed, and cutting depth—were arbitrarily assigned with allowable ranges. The experimental environment was Mori Seiki NVD 1500 DCG for a machine tool, Fanuc 0i for a CNC, System Insights High Speed (average 0.365-s interval) for a power meter, Cold Finish Mild Steel 1018 (10.16 × 10.16 × 1.27 cm) for a workpiece, and a solid carbide flat end mill (8-mm diameter, four flutes) for a cutting tool. The following numbers describe the details of the modeling procedure.

(1) PDA identification: A set of PDA is identified based on the influences of individual data attributes on energy. A machine, workpiece, cutting tool, operation (e.g., contouring, slotting, pocketing, and drilling), command (e.g., G01, G02, and G03), and trajectory (e.g., approach, linear feed, circular feed, retract, stepover, and back) can be MC data attributes, because much research including [55, 59–61] found their influences on energy (the rationales are out of the scope of this work). Here, the data attributes, including the machine, workpiece, and cutting tool, are fixed values, because they are identically used in this experimental environment, whereas operation, command, and trajectory are changeable, depending on each block in an NC program. Feedrate, spindle speed, and cutting depth are set to cause data attributes (cutting width is fixed as the tool diameter) and energy to the effect data attribute. Consequently, PDA can be identified below, and Eq. (4) expresses the numerical function we need to derive:

$$\begin{aligned}
&\text{MC data attributes} = \{\text{operation, command, trajectory}\}\\
&\text{Cause data attributes} = \{\text{feedrate, spindle speed, cutting depth}\}\\
&\text{Effect data attribute} = \{\text{energy}\}\\
&\text{Energy} = f(\text{feedrate, spindle speed, cutting depth}) + \varepsilon
\end{aligned} \tag{4}$$

(2) Data acquisition: Process planning (STEP-NC programs), NC programming (NC programs), and machine-monitoring (MTConnect documents) data are collected in a way that will be described in Section 5.2. STEP-NC programs are used to acquire the MC and cause data attributes, including operation, feedrate, spindle speed, and cutting depth. NC programs are used to acquire the other data attributes, including command and trajectory. MTConnect documents are used to acquire the effect data attribute, energy.

Table 4 List of process parameters

Trial	Feedrate (mm/tooth)	Spindle speed (RPM)	Cutting depth (mm)
1	0.0127	1500	1.5
2	0.0127	2000	1.5
3	0.0127	1750	1
4	0.0229	1750	1
5	0.0127	1750	2
6	0.0178	1500	1
7	0.0178	2000	1
8	0.0178	2000	2
9	0.0178	1750	1.5
10	0.0076	1750	1.5
11	0.0152	1750	1.5
12	0.0127	1750	1.5

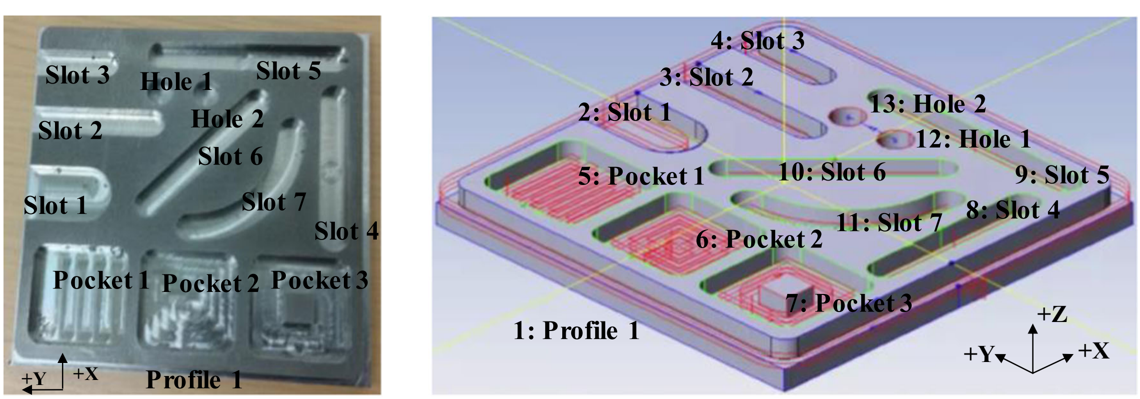


Fig. 12 Test part and tool path trajectories

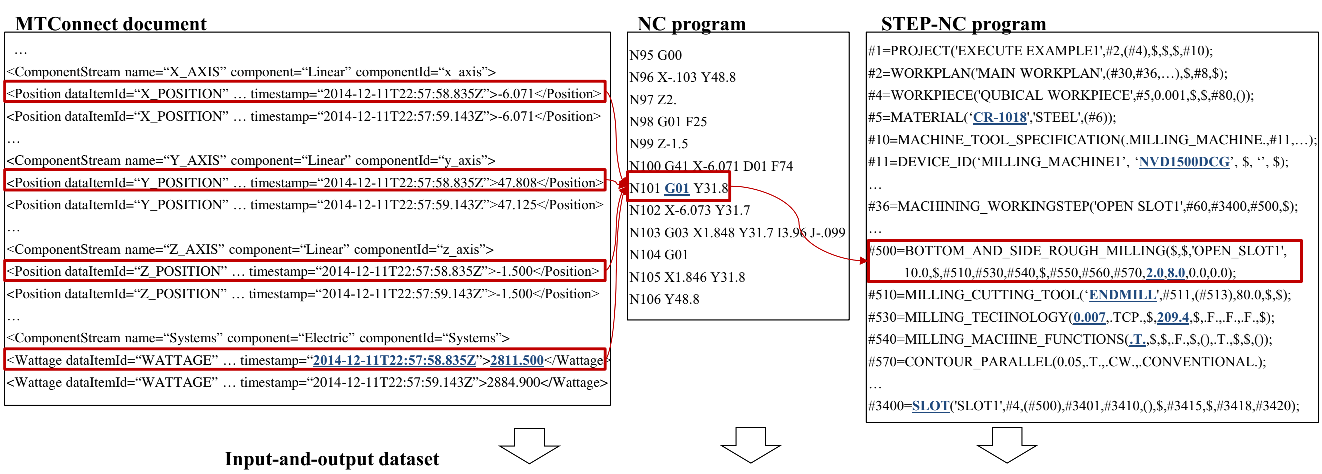
(3) Data pre-processing: Null and less than 1500-watt power values are removed, because they are missing or erroneous data where a minimum of 1500-watt is consumed when the machine is turned on. The uppermost 0.5% and the lowermost 0.5% power values among the entire dataset are eliminated, because these values can be treated as outliers.

(4) Context synchronization: STEP-NC and NC programs and MTConnect documents are synchronized on timestamps. Individual MTConnect data attributes match with their corresponding data attributes in the STEP-NC and NC programs. Figure 13 presents an example. The two wattage values in the MTConnect can be synchronized with the N101 block in the NC program, because their associating positions exist between the start and end points of the N101 block. By inferring from the coordinates and the geometry of the machining feature where the N101 block is executing, the moving direction (-Y) derived from the start and end points indicates that the N101 block is doing feed trajectory. In turn, the N101 block is synchronized with its associating machining operation in the STEP-NC program due to the inference from the feature geometry and the tool path strategy.

(5) Training dataset preparation: The bottom table in Fig. 13 presents an example of training datasets. The first line (red boxed) includes the MC and CE data attributes at a given timestamp. In such a way, multiple training datasets that are separated by the MC data attributes can be derived.

(6) Component model computation: Fifty-one component models that correspond to all the combinations of the MC data attributes given in all the NC programs are created. Here, an ANN technique (two hidden layers and five neurons/layer) is applied using KNIME, a statistics, and data mining tool [62]. For example, the NC blocks for the slotting of Slot 1 in Fig. 12 successively create five component models for approach, linear feed, circular feed, and back and retract trajectories on each cutting layer. Their ANN models vary with commands and trajectories in the experimental environment where the rest of the MC data attributes remain the same.

(7) Model validation: ANN-based energy models make it possible to predict the energy consumed during the execution of an NC program. Table 5 shows the comparison result between the measured and predicted energy. Table 5 first compares the predicted energy values in the case of the DA approach applied (with DA) or not (without DA). In "without DA," energy models without the granularity of models are directly computed from the numerical relationships between sets of process parameters and their total measured values on all the trials using the same ANN technique above. In "with DA," this table then compares statistical significance of two different techniques: ANN- (see Eq. (2)) and second-order polynomial regression (see Eq. (3))-based models. Table 5 includes RMSE as well as total relative error (TRE), which measures the difference between total measured and predicted energy values. The result shows the superiority of the DA applied case than the case without the DA approach. In the two DA cases, it is observable that the ANN-based models make a slightly better performance than the regression-based models due to the



Input-and-output dataset

Timestamp	Manufacturing Configuration Attribute							Input Attribute				Output
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	x_1	x_2	x_3	x_4	y
2014-10-30T21:58:50.718Z	**NVD1500DCG**	**Endmill**	**Wet**	**CR-1018**	**Slotting**	**G01**	**Feed**	**0.007**	**209.4**	**2.0**	**8.0**	**2811.500**
2014-10-30T21:58:50.967Z	NVD1500DCG	Endmill	Wet	CR-1018	Slotting	G01	Feed	0.007	209.4	2.0	8.0	2884.900
…												
2014-11-01T20:37:17.367Z	NVD1500DCG	Endmill	Wet	CR-1018	Slotting	G01	Feed	0.013	157.1	1.5	8.0	2588.8
2014-11-01T20:37:17.568Z	NVD1500DCG	Endmill	Wet	CR-1018	Slotting	G01	Feed	0.013	157.1	1.5	8.0	2586.4
…												

Fig. 13 Example of context synchronization and training dataset preparation

suitability for reducing the model form error ($\varepsilon_{\text{model}}$) when training datasets are learned in this experiment. We conjecture that the proposed approach can provide good predictability because it scales down the granularity of prediction up to specific tool path trajectories.

Compared with the theoretical and experimental approaches mentioned in Section 2.2.3, the proposed approach can enhance practicability. Many of the previous models are unable to predict the energy consumed during the execution of an NC program because they only focused on finding the relationships between process parameters and the total energy (see the result of "without DA" in Table 5). Their focus requires the DOE to investigate the cutting energy, which is affected by process parameters, within a single or a few feed trajectories. Whereas, the proposed approach that excludes the DOE but uses real data can generate component models that predict energy accurately, along with the execution of an NC program.

5.1.2 Machining time prediction

Time prediction modeling was performed in the identical experiment environment specified in Section 5.1.1. The purpose of this experiment was to create time prediction models (hereafter, time models) to forecast the machining time spent during the execution of an NC program for fabricating the same part visualized in Fig. 12. We use the same 12 sets of process parameters listed in Table 4. The differences with the energy modeling explained in Section 5.1.1 are as follows:

(1) PDA identification: The effect data attribute is set to time because the machining time is the indicator focused on in this experiment. PDA can be identified below, and Eq. (5) expresses the numerical function we need to derive:

$$\begin{aligned}&\text{MC data attributes} = \{\text{operation}, \text{command}, \text{trajectory}\}\\&\text{Cause data attributes} = \{\text{feedrate}, \text{spindle speed}, \text{cutting depth}\}\\&\text{Effect data attribute} = \{\text{time}\}\\&\text{Time} = f(\text{feedrate},\ \text{spindle speed},\ \text{cutting depth}) + \varepsilon\end{aligned} \tag{5}$$

(2) Data acquisition: MTConnect documents are used to acquire the effect data attribute—machining time. The machining time value on each trajectory can be obtained by subtracting the starting time from the finishing time on the identical trajectory.

(3) Data pre-processing: The datasets that contain missing or incomplete data are removed. For example, we remove the data tuples that exist between the data tuple starting with a feed trajectory and the data tuple ending with a back trajectory. These data have a high possibility of including missing data in-between the starting and finishing data tuples.

(4) Context synchronization: Timestamps in MTConnect documents are used to calculate the starting and finishing time values on a trajectory. We can match the location coordinate on each timestamp with its associated NC block in the same manner described in Section 5.1.1. Because the first timestamp or the last timestamp that

Table 5 Comparison result between measured and predicted energy

Trial	Measured energy (kJ)	Without DA		With DA					
				ANN			Regression		
		Predicted energy (kJ)	TRE (%)	Predicted energy (kJ)	TRE (%)	RMSE (J)	Predicted energy (kJ)	TRE (%)	RMSE (J)
1	13,592.5	14,267.4	2.26	13,950.3	−0.02	28.29	13,901.1	−0.37	28.67
2	11,382.1	11,309.5	−0.64	11,385.4	0.03	29.41	11,414.2	0.28	29.72
3	19,535.2	18,808.2	−3.72	19,530.0	−0.03	23.34	19,457.8	−0.40	23.62
4	9830.3	9570.3	−2.65	9834.1	0.04	28.29	9823.5	−0.07	28.39
5	9943.1	9763.8	−1.80	9967.9	0.25	34.03	10,007.9	0.65	34.50
6	13,365.9	13,300.4	−0.49	13,371.7	0.04	24.90	13,417.1	0.38	25.30
7	11,044.0	11,300.0	2.32	11,045.8	0.02	25.92	11,076.5	0.30	26.26
8	6012.6	7093.1	17.97	5978.5	−0.57	40.40	5947.6	−1.08	41.10
9	9750.7	9473.6	−2.84	9741.2	−0.10	32.31	9720.8	−0.31	32.58
10	19,281.6	18,916.3	−1.89	19,292.8	0.06	21.82	19,298.4	0.09	21.82
11	10,791.6	10,360.8	−3.99	10,847.7	0.52	29.55	10,873.6	0.76	30.12
12	12,580.1	12,725.1	1.15	12,522.2	−0.46	29.97	12,529.8	−0.40	29.98

appeared in an NC block corresponds to the starting time or the finishing time for the NC block, respectively, machining time can be calculated from subtracting the starting time from the finishing time. In such a way, the context synchronization for machining time can be performed to make training datasets.

(5) Component model computation: Fifty-one component models (the same number of component models as in Section 5.1.1) are created. Similarly with energy prediction (Section 5.1.1), the ANN and second-order polynomial techniques are applied using KNIME.

(6) Model validation: Both ANN and regression-based models can predict the machining time consumed during the execution of an NC program. Table 6 also shows the statistical comparison between the measured and predicted time values. The result shows that the two techniques used for the DA approach predict machining time more accurately, and the ANN-based models score better TRE values (the less, the better) in nine trials among the 12 trials. The proposed approach can also result in good predictability in another application, machining time prediction.

5.2 Prototype of data warehouse

A prototype of the data warehouse was implemented based on the methods presented in Section 4.2. This warehouse consists of two sub-warehouses, (a) Static Data Warehouse (SDW) and (b) Dynamic Data Warehouse (DDW), as shown in Fig. 14. The SDW imports and exports static design/planning data through the connections to existing manufacturing databases, while the DDW processes dynamic data coming from data acquisition systems during machining. The reason for splitting into two different warehouses is the nature of the difference between static and dynamic data. The static data (e.g., resource description, process planning, and NC programming) are typically time independent, structured, and, often, average-sized documents, most of which have schemas that define their syntax or structure. The dynamic data (e.g., machine-monitoring, measurement, and quality), however, are time series and are not structured, and each data size is small but generated in large quantities in a short time. In other words, while the static data are rarely inserted and retrieved of relatively large amounts of data, the dynamic data insert very small amounts of data at very high speed. Also, when retrieving the dynamic data, it is necessary to find small data among large sets of data.

For handling the static data, we implemented a Plain Old Java Object (POJO) model mapper based on the document schema, as explained in Section 4.2. It enabled the documents to be serialized and converted seamlessly to simple JSON scripts [63]. For example, a STEP-NC part program can be converted to a JSON-formatted file using this mapper. Whereas, we implemented a message queue as well as another POJO model for dynamic data handling. The message queue is a queue of data messages sent from machines to the DDW. It enables high-speed streaming communication between multiple machines and the DDW through sequential data processing. The below items are the details of operations in these two sub-warehouses. We use STEP-NC and NC programs as examples of the static data and MTConnect documents as an

Table 6 Comparison result between measured and predicted machining time

Trial	Measured time (s)	Without DA		With DA					
				ANN			Regression		
		Predicted time (s)	TRE (%)	Predicted time (s)	TRE (%)	RMSE (ms)	Predicted time (s)	TRE (%)	RMSE (ms)
1	4673.9	4710.3	0.78	4671.2	−0.06	107.36	4760.2	1.85	107.51
2	3821.4	3902.7	2.13	3835.9	0.38	117.19	3845.5	0.63	117.21
3	6762.1	6693.0	−1.02	6771.7	0.14	23.88	6835.4	1.08	24.09
4	3580.9	3539.2	−1.16	3583.6	0.08	13.45	3619.8	1.09	13.84
5	3380.9	3381.7	0.02	3393.8	0.38	17.35	3380.3	−0.02	17.26
6	4932.5	4952.2	0.40	4927.7	−0.10	105.73	4843.6	−1.80	105.89
7	4027.5	4047.4	0.49	4022.0	−0.14	113.15	4001.2	−0.65	113.17
8	2063.6	2063.8	0.01	2019.6	−2.13	18.70	2063.5	−0.01	17.39
9	3279.1	3289.9	0.33	3291.0	0.36	18.09	3286.9	0.24	18.06
10	6756.2	6713.7	−0.63	6730.5	−0.38	124.05	6709.9	−0.68	124.06
11	3697.7	3624.1	−1.99	3699.7	0.06	20.82	3662.5	−0.95	21.00
12	4293.7	4230.1	−1.48	4307.7	0.32	110.64	4249.4	−1.03	110.69

example of the dynamic data for the data acquisition described in Section 5.1.

(a) SDW, see Fig. 14a: The format adaptor in the data storing service parses the data instances formatted by the original data representations (e.g., an EXPRESS file for a STEP-NC part program). The JSON translator translates and forms these data instances into the format of JSON objects under compliance with POJO model mappers, which predefined the mapping rules between the JSON scheme and those original schemas of the STEP-NC and NC programs. MongoDB, a distributed database management system [40], stores JSON-formatted data instances in multiple data nodes. HDFS, which is designed to store very large datasets and to stream those datasets at high bandwidth [39], and HBase generates and manages the metadata that provide searching and locating information about data instances. MySQL stores the metadata and helps retrieve JSON objects from MongoDB. Hibernate not only creates query languages but also retrieves and maps JSON objects into tuples of MongoDB [64]. The format adaptor in the data retrieving service transforms the tuples of the returned JSON objects into designated data formats, such as a comma-separated values file.

(b) DDW, see Fig. 14b: Format adaptors parse the data instances given through MTConnect agents. RabbitMQ is used to make queues and allocate keys for identifying data instances received from multiple machines [65]. The JSON translator translates and forms these data instances in the format of JSON, which is a similar structure to MongoDB documentation. Spark provides application programming interfaces for large-scale data processing [66]. MapReduce, which defines a programming model for processing and generating large datasets [67], supports the creation of large-scale datasets for responding to query invoking.

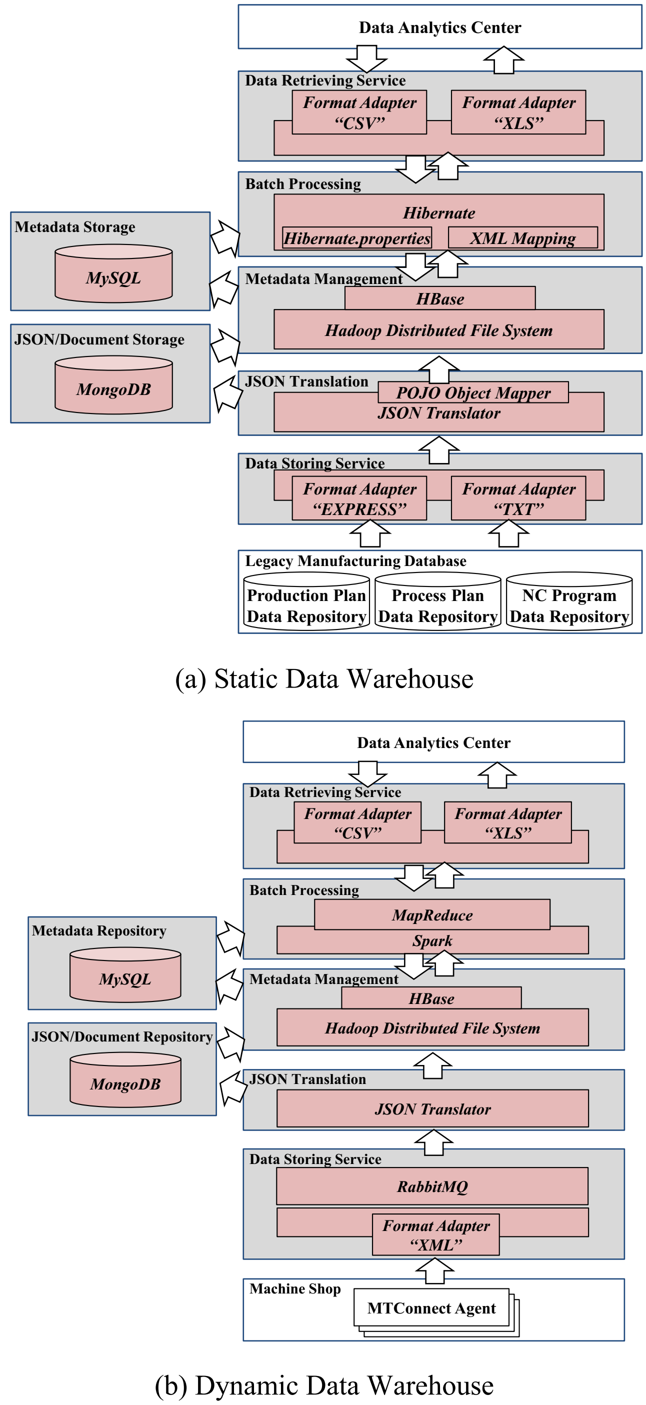


Fig. 14 **a**, **b** Implementation architecture of data warehouse

The performance of the data warehouse prototype was measured. We measured the inserting performance in SDW and DDW in regard to the increase in the number of records inserted. The insert transaction is expected to be performed most frequently in a typical manufacturing environment, especially high-speed processing. The test environment is Xeon ×5460 for CPU, 8 GB for RAM, StorageTek Controller (256 MB, write-back) for Storage, and 4 × 10 K SAS/RAID 0 for hard drive.

Figure 15a shows the insert performance comparison between SDW and DDW. As the amount of data inserted increases, the DDW processes at a relatively constant rate, whereas the SDW demonstrates a drastically slower processing speed. That is, it is observable that the DDW is superior for high-speed data processing. We also measure query performance in SDW and DDW. The query transaction typically takes the longest time among insert, query, update, and delete transactions in relational database systems. The test was performed in the same experimental environment, and 10% of the inserted data were retrieved. Figure 15b shows the query performance comparison between the SDW and DDW. The SDW gives slightly better results than the DDW. This is thought to be an efficient search using the metadata in

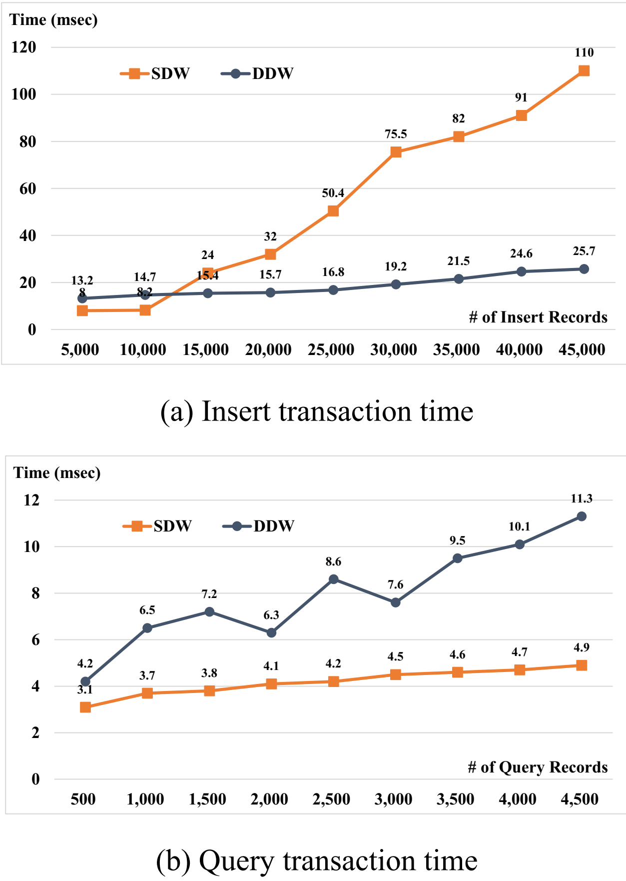


Fig. 15 **a**, **b** Performance comparison between SDW and DDW

structured documents. These performance measurements imply that the two different sub-warehouses cause optimal performance in their heterogeneous natures and result in experimental rationales for splitting them into two sub-warehouses.

5.3 Prototype of virtual shop floor

A prototype of the virtual shop floor was implemented based on the methods described in Section 4.1. The implementation environment was Eclipse Neon for computer programing, and JADE for agent-based software development and deployment [49].

Figure 16 shows an implementation scenario. A given scenario is that an order agent (order 1) tries to find the best energy-efficient machine among three machine agents (machines 1, 2, and 3) after a product agent (product 1) requests a milling machining task from the order agent. Once the order agent announces the task, each machine receives relevant product and task information (see Section 5.2) via a data broker agent (data broker 1) with the connection of the data warehouse. It then requests associated energy models (see Section 5.1) from a model broker agent (model broker 1). The model broker links with the data analytics model pool for searching energy models and returns the associated energy models to the individual machine agents. Here, process parameters given in three machines are as follows: machine 1 (feedrate: 0.0127 mm/tooth, spindle speed: 1500 RPM, cutting depth: 1.5 mm), machine 2 (0.0178, 2000, 2.0, respectively), and machine 3 (0.0127, 1750, 1.0, respectively). Each set of process parameters, respectively, associates with the set of process parameters given in trial 1, trial 8, and trial 3 in Table 4. These machine agents calculate anticipated energy using the received energy models and propose their bids for taking the task. The order agent accepts the best proposal that suggests the minimum energy demand and informs this bid result to the product agent. The selected machine is assigned to process the given task.

Figure 17 shows a screen shot of the prototype to implement the scenario in Fig. 16 and visualizes automatic message interactions between agents without human intervention. Each agent is registered on JADE DF to connect with the Web service in the initial stage. The three machine agents retrieve associated energy models via model broker 1, and they calculate their respective energy values as follows: MACHINE 1 (13,950.3 kJ), machine 2 (5978.5 kJ), and machine 3 (19,530.0 kJ). Finally, machine 2 is accepted from order 1, because it submits the minimum energy value. These energy values, respectively, match with the predicted energy values in trial 1, trial 8, and trial 3 in Table 5.

6 Conclusion

We presented the design and implementation of a BDA platform in manufacturing to achieve intelligent, autonomous, and collaborative decision-making and seamless data exchange, which were addressed as the three major characteristics in Section 1. The proposed platform focuses on embodying the three implementation challenges—*object virtualization*, *data control*, and *model control*—that traditional platforms have hardly covered. For these purposes, we analyzed design requirements, designed a system architecture for the platform, developed technical methods for running the operational mechanism in the platform, and implemented a prototype system for demonstrating the feasibility and effectiveness of the platform.

It is currently known that CPPS should ensure data streamlining from the physical space to cyber space, information feedback from the cyber space, and data analytics that constructs the cyber space; however, these requirements are not specific enough for implementation purpose [9, 13]. The present work dealt with such requirements in terms of *data control*, *object virtualization*, and *model control*, respectively, and introduced their technical solutions with their prototype implementation. Thus, we expect the proposed platform can be a referential architecture that accommodates BDA and CPPS.

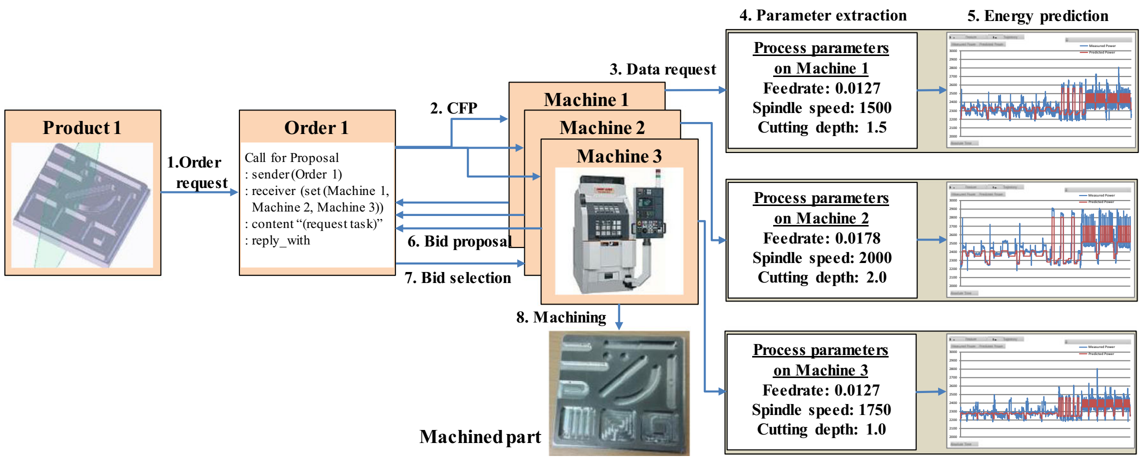


Fig. 16 Implementation scenario

In designing the proposed platform, we strived to increasing generality of the platform and interoperability with other manufacturing platforms through the utilization of de jure and de facto standards. Data integration using MT standards such as STEP-NC and MTConnect and ICT standards like JSON was considered to facilitate seamless data exchange across multiple platforms. Systems integration using JADE and FIPA was taken into account for open interaction between heterogeneous applications. These integration capabilities expectedly make the proposed platform interoperable with general manufacturing platforms.

The completion of implementing the proposed platform helps manufacturers (e.g., machine tool builders, application developers, and process engineers) gain manufacturing intelligence through the use of models specific for 3M, implement a cost-effective environment especially for small-and-medium-sized enterprises through the use of standards and open-source solutions, obtain a technical reference for implementing a data analytics environment time-efficiently, and, eventually, increase productivity and sustainability performance.

However, some issues remain. The present work excluded the implementation of optimization and disturbance handling, which directly relate to increasing productivity and sustainability performance on shop floors. The present work also excluded UQ integration, which should be accompanied with

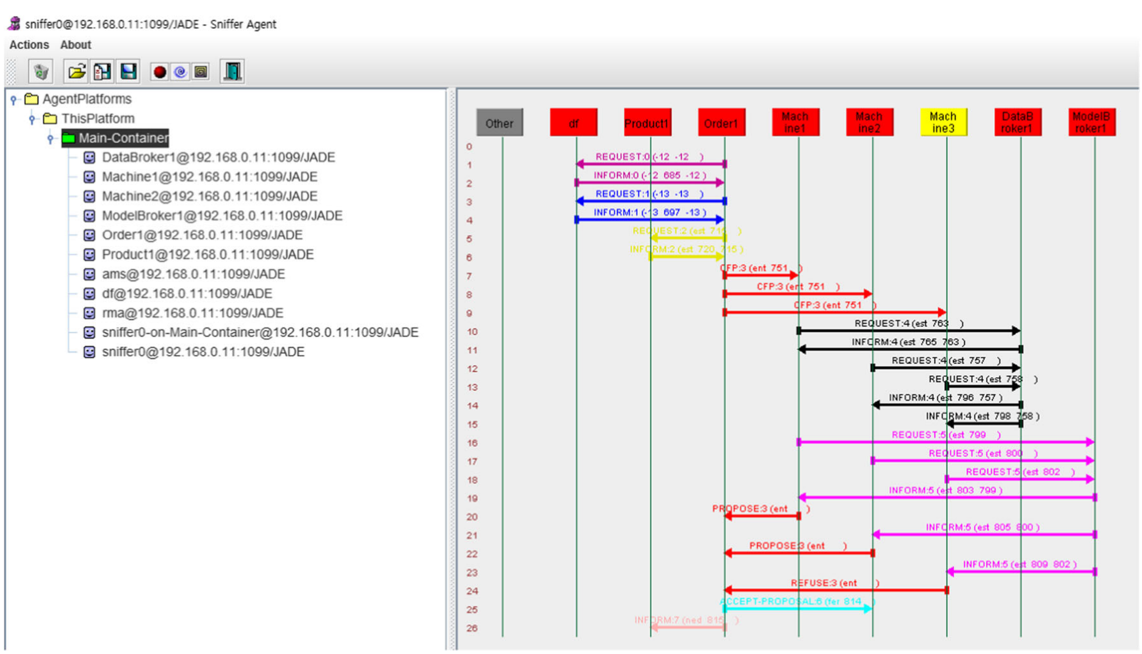


Fig. 17 Screen shot of the virtual shop floor prototype

data-driven modeling. The proposed platform is under construction to raise the degree of completion, which requires assistant module implementation and module integration, although basic functionalities of the platform have been implemented. We plan to address these issues in future research. Furthermore, we plan to advance a real-time CPPS environment where real-time control will be more challenging.

Funding This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2018R1D1A1B07047100).

References

1. Davis J, Edgar T, Porter J, Bernaden J, Sarli M (2012) Smart manufacturing, manufacturing intelligence and demand-dynamic performance. Comput Chem Eng 47:145–156
2. Kang HS, Lee JY, Choi SS, Kim H, Park JH, Son JY, Kim BH, Noh SD (2016) Smart manufacturing: past research, present findings, and future directions. Int J Pr Eng Man-GT 3(1):111–128
3. Smart Manufacturing Leadership Coalition (2011) Implementing 21st century smart manufacturing—workshop summary report. https://smartmanufacturingcoalition.org. Accessed 10 May 2016
4. Monostori L, Kadar B, Bauernhansl T, Kondoh S, Kumara SRT, Reinhart G, Sauer O, Schuh G, Sihn W, Ueda K (2016) Cyber-physical systems in manufacturing. CIRP Ann Manuf Technol 65: 621–641
5. Brussel HV, Wyns J, Valckenaers P, Bongaerts L, Peeters P (1998) Reference architecture for holonic manufacturing systems: PROSA. Comput Ind 37:255–274
6. Suh SH, Shin SJ, Yoon JS, Um JM (2008) UbiDM: a new paradigm for product design and manufacturing via ubiquitous computing technology. Int J Comput Integr Manuf 21(5):540–549
7. Zuehlke D (2010) SmartFactory – towards a factory-of-things. Annu Rev Control 34:129–138
8. Babiceanua RF, Sekerb R (2016) Big data and virtualization for manufacturing cyber-physical systems: a survey of the current status and future outlook. Comput Ind 81:128–137
9. Lee J, Bagheri B, Kao HA (2015) A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manuf Lett 3:18–23
10. Rajkumar R, Lee IS, Sha L, Stankovic J (2010) Cyber-physical systems: the next computing revolution, 47th ACM/IEEE Design Automation Conference, 731–736, Anaheim, CA, U.S.A.
11. Bi Z, Cochran D (2014) Big data analytics with applications. Journal of Management Analytics 1(4):249–265
12. Shin SJ, Woo JY, Rachuri S (2014) Predictive analytics model for power consumption in manufacturing. Proc CIRP 15:153–158
13. Lee J, Ardakani HD, Yang S, Bagheri B (2015) Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. Proc CIRP 38:3–7
14. Aerts ATM, Goossenaerts JBM, Hammer DK, Wortmann JC (2004) Architectures in context: on the evolution of business, application software, and ICT platform architectures. Inf Manag 41: 781–794
15. Valilai OF, Houshmand M (2013) A collaborative and integrated platform to support distributed manufacturing system using a service-oriented approach based on cloud computing paradigm. Robot Comput Integr Manuf 29:110–127
16. Valilai OF, Houshmand M (2010) INFELT STEP: an integrated and interoperable platform for collaborative CAD/CAPP/CAM/CNC machining systems based on STEP standard. Int J Comput Integr Manuf 23(12):1095–1117
17. Wang HF, Zhang YL (2002) CAD/CAM integrated system in collaborative development environment. Robot Comput Integr Manuf 18:135–145
18. Nylund H, Andersson PH (2010) Simulation of service-oriented and distributed manufacturing systems. Robot Comput Integr Manuf 26:622–628
19. Wang L (2011) Planning towards enhanced adaptability in digital manufacturing. Int J Comput Integr Manuf 24(5):378–390
20. Leitao P (2009) Agent-based distributed manufacturing control: a state-of-the-art survey. Eng Appl Artif Intell 22:979–991
21. Monostori L, Vancza J, Kumara SRT (2006) Agent-based systems for manufacturing. Ann CIRP 55:697–720
22. Colombo AW, Schoop R, Neubert R (2006) An agent-based intelligent control platform for industrial holonic manufacturing systems. IEEE Trans Ind Electron 53(1):322–337
23. Oztemel E, Tekez EK (2009) A general framework of a Reference Model for Intelligent Integrated Manufacturing Systems (REMIMS). Eng Appl Artif Intell 22:855–864
24. Yin JW, Zhang WY, Cai M (2010) Weaving an agent-based semantic grid for distributed collaborative manufacturing. Int J Prod Res 48(7):2109–2126
25. Lin J, Long Q (2011) Development of a multi-agent-based distributed simulation platform for semiconductor manufacturing. Expert Syst Appl 38:5231–5239
26. Mikos WL, Ferreira JCE, Botura PEA, Freitas LS (2011) A system for distributed sharing and reuse of design and manufacturing knowledge in the PFMEA domain using a description logics-based ontology. J Manuf Syst 30:133–143
27. Nassehi A, Newman ST, Allen RD (2006) The application of multi-agent systems for STEP-NC computer aided process planning of prismatic components. Int J Mach Tool Manu 46:559–574
28. Newman ST, Nassehi A (2007) Universal manufacturing platform for CNC machining. Ann CIRP 56:459–462
29. Xu X (2009) Integrating advanced computer-aided design, manufacturing, and numerical control: principles and implementations. Information Science Reference, New York
30. Valilai OF, Houshmand M (2011) LAYMOD; a layered and modular platform for CAx collaboration management and supporting product data integration based on STEP standard. International Journal of Computer, Electrical, Automation, Control and Information Engineering 5(6):633–641
31. Xu X (2012) From cloud computing to cloud manufacturing. Robot Comput Integr Manuf 28:75–86
32. Wang XV, Xu XW (2013) An interoperable solution for Cloud manufacturing. Robot Comput Integr Manuf 29:232–247
33. Huang B, Li C, Yin C, Zhao X (2013) Cloud manufacturing service platform for small- and medium-sized enterprises. Int J Adv Manuf Technol 65:1261–1272
34. Song T, Liu H, Wei C, Zhang C (2014) Common engines of cloud manufacturing service platform for SMEs. Int J Adv Manuf Technol 73:557–569
35. Helo P, Suorsa M, Hao Y, Anussornnitisarn P (2014) Toward a cloud-based manufacturing execution system for distributed manufacturing. Comput Ind 65:646–656
36. Shen W, Hao Q, Yoon HJ, Norrie DH (2006) Applications of agent-based systems in intelligent manufacturing: an updated review. Adv Eng Inform 20:415–431
37. MTConnect Institute (2014) MTConnect Standard Part 1 – Overview and Protocol version 1.3.0. The Association for Manufacturing Technology, McLean

38. Pavlo A, Paulson E, Rasin A, Abadi DJ, DeWitt DJ, Madden S, Stonebraker M (2009) A comparison of approaches to large-scale data analysis. Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, June 29 – July 2, Rhode Island, U.S.A.
39. Shvachko K, Kuang H, Radia S, Chansler R (2010) The Hadoop Distributed File System. 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, May 3–7, Incline Village, U.S.A.
40. MongoDB. www.mongodb.org. Accessed 03 June 2015
41. HBase. http://hbase.apache.org/book.html#arch.overview. Accessed 10 October 2016
42. Xu X, Wang L, Newman ST (2011) Computer-aided process planning—a critical review of recent developments and future trends. Int J Comput Integr Manuf 24(1):1–31
43. Kara S, Li W (2011) Unit process energy consumption models for material removal processes, CIRP annals—manufacturing technology 60:37–40
44. Dean EB (2000) Design of experiments from the perspective of competitive advantage. http://spartan.ac.brocku.ca/~pscarbrough/dfca1stmods/dfc/doe.html. Accessed 7 February 2017
45. Wazed MA, Ahmed S, Yusoff N (2009) Uncertainty factors in real manufacturing environment. Aust J Basic Appl Sci 3(2):342–351
46. Nannapaneni S, Mahadevan S, Rachuri S (2016) Performance evaluation of a manufacturing process under uncertainty using Bayesian networks. J Clean Prod 113:947–959
47. Babiceanu RF, Chen FF (2006) Development and applications of holonic manufacturing systems: a survey. J Intell Manuf 17:111–131
48. Poslad S (2007) Specifying protocols for multi-agent systems interaction. ACM Trans Auton Adap 2(4):15–es
49. Bellifemine F, Caire G, Greenwood D (2006) Developing multi-agent systems with JADE. John Wiley & Sons, Ltd., Chichester
50. Smith RG (1980) The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans Comput 29(12):1104–1113
51. OASIS (2004), UDDI specification technical committee draft. http://www.uddi.org/pubs/uddi-v3.0.2-20041019.htm. Accessed 20 November 2016
52. ECMA International (2013) ECMA—404: The JSON data interchange format. https://www.ecma-international.org/publications/standards/Ecma-404.htm. Accessed 10 February 2015
53. CRISP-DM Consortium (2000) Cross industry standard process for data mining—step-by-step data mining guide. https://www.the-modeling-agency.com/crisp-dm.pdf. Accessed 29 November 2016
54. Kotsiantis SB, Kanellopoulos D, Pintelas PE (2006) Data preprocessing for supervised learning. Int J Comput Sci 1(2):111–117
55. Peng T, Xu X (2014) Energy-efficient machining systems: a critical review. Int J Adv Manuf Technol 72:1389–1406
56. Xu W, Cao L (2014) Energy efficiency analysis of machine tools with periodic maintenance. Int J Prod Res 52(18):5273–5285
57. Liang B, Mahadevan S (2011) Error and uncertainty quantification and sensitivity analysis in mechanics computational models. Int J Uncertain Quantif 1(2):147–161
58. Guazzelli A, Zeller M, Lin WC, Williams G (2009) PMML: an open standard for sharing models. R J 1(1):60–65
59. Aramcharoen A, Mativenga PT (2014) Critical factors in energy demand modelling for CNC milling and impact of toolpath strategy. J Clean Prod 78:63–74
60. Mulyadi IH, Balogun VA, Mativenga PT (2015) Environmental performance evaluation of different cutting environments when milling H13 tool steel. J Clean Prod 108:110–120
61. Shin SJ, Woo JY, Rachuri S (2017) Energy efficiency of milling machining: component modeling and online optimization of cutting parameters. J Clean Prod 161:12–29
62. Berthold MR, Cebron N, Dill F, Gabriel TR, Kotter T, Meinl T, Ohl P, Thiel K, Wiswedel B (2009) KNIME—the Kostanz information miner. SIGKDD Explor 11(1):26–31
63. Brian SB (2006) Beginning POJOs: lightweight Java web development using plain old Java objects in spring, hibernate, and tapestry. Apress, New York
64. Hibernate. http://hibernate.org/orm/. Accessed 10 October 2016
65. RabbitMQ. https://www.rabbitmq.com/features.html. Accessed 10 October 2016
66. Spark. http://spark.apache.org/docs/latest/sql-programming-guide.html. Accessed 10 October 10 2016
67. Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113