https://doi.org/10.1016/j.rcim.2021.102217

Service-oriented Industrial Internet of Things Gateway for Cloud Manufacturing

Chao Liu+

State Key Laboratory of Ultra-precision Machining Technology,

Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China

+Equal contribution

Ziwei Su+

Department of Mechanical Engineering, The University of Auckland

Auckland, New Zealand, 1010

+Equal contribution

Xun Xu

Department of Mechanical Engineering, The University of Auckland

Auckland, New Zealand, 1010

Yuqian Lu*

Department of Mechanical Engineering, The University of Auckland, New Zealand

OCRID: 0000-0001-5954-0421

Tel.: +64 9 923 1584

E-mail address: yuqian.lu@auckland.ac.nz

Corresponding author

Service-oriented IIoT gateway for Cloud Manufacturing

Abstract

Cloud manufacturing represents a service-oriented manufacturing paradigm that allows ubiquitous and on-demand access to various customisable manufacturing services in the cloud. While a vast amount of research in cloud manufacturing has focused on high-level decision-making tasks, such as service composition and scheduling, the link between field-level manufacturing data and the cloud manufacturing platform has not been well established. Efficient data acquisition, communication, storage, query, and analysis of field-level manufacturing equipment remain a significant challenge that hinders the development of cloud manufacturing systems. Therefore, this paper investigates the implementation of the emerging Industrial Internet of Things (IIoT) technologies in a cloud manufacturing system to address this challenge. We propose a service-oriented plug-and-play (PnP) IIoT gateway solution based on a generic system architecture of IIoT-supported cloud manufacturing system. Service-oriented data schemas for manufacturing equipment are developed to capture just-enough information about field-level manufacturing equipment and allow efficient data storage and query in a cloud time-series database (TSDB). We tested the feasibility and advantages of the proposed approach via the practical implementation of the IIoT gateways on a 3D printer and a machine tool. Our research suggests that purposely developed service-oriented data schemas that capture the essential information for high-level cloud manufacturing decision-making via PnP IIoT technologies are a good solution for connecting field-level manufacturing equipment to a cloud manufacturing platform.

Keywords: cloud manufacturing; industrial internet of things; gateway; service-oriented information model; resource virtualisation

1. Introduction

Cloud manufacturing represents a service-oriented manufacturing paradigm where manufacturing resources are virtualised as manufacturing services that can be managed and configured in an intelligent and unified way in the cloud and allows ubiquitous and on-demand network access [1–3]. Cloud manufacturing aims to achieve efficient integration and sharing of distributed manufacturing resources to enable efficient on-demand production of highly customised products. As a promising trend of the future of manufacturing, ever since the introduction of the concept of cloud manufacturing in 2010 [4], various research topics related to cloud manufacturing such as architecture design, resource virtualisation, service selection and composition, service searching and matching, and task scheduling have been widely discussed and studied in both academia and industry during the last decade [5]. However, despite the vast amount of research effort in this field, one has to admit that the envisioned cloud manufacturing paradigm has not been achieved yet.

Resource virtualisation is a core technology in cloud manufacturing that allows various field-level manufacturing resources (machines, robotics, workpieces, software, knowledge, etc.) to be virtualised as services in the cloud [6,7]. Resource virtualisation functions as a link between the physical manufacturing resources and the cloud, which builds the foundation of all the high-level decision-making tasks such as service composition, selection, matching, and scheduling. According to Lu et al. [7], resource virtualisation of the field-level manufacturing equipment (machine tools, 3D printers, robotics, etc.) requires both the static manufacturing capability information and the dynamic availability information that is reflected by their real-time status. Hence, a critical prerequisite for manufacturing resource virtualisation is the data acquisition of manufacturing equipment. While a significant amount of research works has been conducted on resource capability virtualisation and high-level decision-making tasks in cloud manufacturing, research related to resource availability virtualisation that considers real-time data acquisition of field-level manufacturing equipment and communication between the field-level manufacturing equipment and the cloud manufacturing platform has not been widely investigated. The high diversity of manufacturing equipment, data sources, data formats, and communication protocols have all posed a significant challenge for efficient data acquisition and communication in the context of cloud manufacturing.

Recent advancements in Industrial Internet of Things (IIoT) have shown a great potential of addressing this challenge. IIoT adopts the Internet of Things (IoT) technologies in the industrial environment to achieve flexible and scalable industrial and machine-to-machine (M2M) communications [8]. As demonstrated by the prior research [9], IIoT technologies could be used to capture machine data and streamline them to external business systems and making manufacturing equipment as a service. However, how to utilise IIoT technologies to support cloud manufacturing needs to be further investigated. Firstly, a generic system architecture of IIoT-supported cloud manufacturing system needs to be developed as a strategic implementation guideline. Secondly, the mechanisms of IIoT-enabled data acquisition, communication, storage, query, and analysis of common manufacturing equipment (such as 3D printers and machine tools) in the context of cloud manufacturing need to be investigated. Furthermore, cloud manufacturing's unique issues such as service-oriented manufacturing resource virtualisation, data interoperability between physical level and cloud platform, and data security and privacy also need to be addressed.

In light of the issues above, this paper investigates the implementation of IIoT technologies in a cloud manufacturing environment. Specifically, we propose a service-oriented, plug-and-play (PnP) IIoT gateway solution to facilitate the data acquisition, communication, storage, query, and analysis between field-level manufacturing equipment and a cloud manufacturing platform. The rest of this paper is organised as follows. Section 2 reviews previous works on cloud manufacturing and IIoT and identifies the research gaps. Section 3 introduces the details of the proposed service-oriented IIoT gateway solution, including the system architecture of IIoT-supported cloud manufacturing system, the mechanism of the IIoT gateway, the service-oriented data schemas for manufacturing equipment, the cloud TSDB, and the data transformation process. Section 4 demonstrates the practical development of two IIoT gateways for a 3D printer and a machine tool to validate the feasibility and advantages of the proposed approach. Section 5 concludes the paper and discusses future research works.

2. Literature Review

This section summarises state-of-the-art research on cloud manufacturing and IIoT, and concludes the research gaps that motivated this research work.

2.1 Cloud manufacturing

Manufacturing resource virtualisation is the foundation of all the high-level decision-making activities in cloud manufacturing. Among the various types of manufacturing resources, manufacturing equipment plays a unique role since their static capability and dynamic availability both affect their feasibility as a service. Previous research on manufacturing resource virtualisation mainly focuses on the relatively static manufacturing capability. Ontology has been commonly used to represent the manufacturing capability due to its advantage of semantic interoperability. For example, Lu et al. [10] introduced an ontology-based approach to virtualising manufacturing resource by utilising existing industry standards such as STEP and STEP-NC. Systematic guidance on developing ontologies for manufacturing resources was presented. Liu et al. [11] proposed a multi-granularity resource virtualisation model based on an ontology that considers the workflow, activity, and resource as the main factors. Resource aggregation and clustering algorithms were applied to achieve the virtualisation process. Luo et al. [12] developed a multidimensional information model and an ontology-based description method to realise the virtualisation of manufacturing capability in cloud manufacturing. Resource virtualisation of dynamic manufacturing availability that considers real-time data acquisition and communication in cloud manufacturing has been rarely investigated, although the use of IoT and Cyber-physical System (CPS) technologies for field-level manufacturing data collection has been discussed in some conceptual cloud manufacturing frameworks, such as the ones proposed in [13] and [14].

Service composition and selection is a fundamental issue of on-demand manufacturing service provision in cloud manufacturing [15]. Quality of service (QoS) is usually used as a critical indicator of the performance of a service. The commonly used QoS properties for manufacturing equipment include lead time, cost, reliability, availability, and maintainability. Lartigau et al. [16]

proposed a service composition method based on QoS evaluation while considering the geo-perspective transportation constraints introduced by the locations of manufacturing resources. To solve the problem of multitask corresponding multi-service selection, Yuan et al. [17] developed a mathematical service composition model considering six QoS properties, including time, composability, quality, usability, reliability, and cost. The proposed method was verified in a complex mold manufacturing case study. Since semantic web allows efficient query and reasoning of knowledge, it has also been applied to perform the service composition tasks. Lu and Xu [18] proposed a semantic web-based framework for service composition in cloud manufacturing. A practical web-based system was developed to distribute engineering knowledge of manufacturing resources for intelligent service composition and adaptive resource planning. Based on the service arrangement and the QoS of each subtask, Yang et al. [19] proposed a robust service composition and optimal selection (rSCOS) method for cloud manufacturing. A guiding artificial bee colony – grey wolf optimisation algorithm was developed to improve the robustness of the rSCOS process. However, their experiments used simulated data without considering practical data acquisition issues. Recently, Liang et al. [20] introduced a deep reinforcement learning-based approach to solving the QoS-aware service composition problem in cloud manufacturing with the consideration of logistics. The Deep Q-Network, the dueling architecture, and the prioritized replay mechanism were integrated to achieve efficient and scalable service composition. Yu et al. [21] also proposed a blockchain-based cloud manufacturing system in which blockchain was used to intermediate the service composition and record transaction results.

Task and service scheduling is another research focus in cloud manufacturing since it determines the efficiency of resource allocation and the effectiveness of service utilisation [22]. Though scheduling in cloud manufacturing relies closely on the real-time perception of field-level manufacturing equipment, previous research on scheduling focused mainly on the scheduling algorithms and decision-making processes while usually assuming the real-time status of manufacturing equipment is already available in the cloud manufacturing platform. Li et al. [23] proposed a scheduling model for distributed robots to achieve cooperative task execution in a cloud manufacturing environment. Four robot deployment methods were developed and tested in a simulation experiment. Liu et al. [24] introduced a multitask scheduling model for cloud manufacturing that incorporates task workload modelling while considering service quantity, service efficiency, and enterprise capacity. In their simulation experiment, the availability of manufacturing equipment was again assumed in the cloud manufacturing platform. Recently, Wang et al. [25] proposed an advanced planning and scheduling system framework in the context of cloud manufacturing. Package diagram was used to improve modeling efficiency and data stability. The proposed system was deployed in the Amazon Web Services (AWS) cloud platform and validated though a real-world scheduling task in the printed circuit board production.

2.2 Industrial Internet of Things

The rapid development of Information and Communication Technology (ICT) has triggered the introduction of IIoT, where smart objects, cyber-physical assets, information technologies, and cloud/edge computing platforms are networked to enable real-time, intelligent, and autonomous access, collection, analysis, communications, and exchange of process, product, and service information within the industrial environment [26]. In recent years, the great benefits of IIoT have attracted significant attention from both industry and academia. Based on the report of Gartner magic quadrant for IIoT platforms [27], currently, a considerable amount of commercial IIoT platform solutions have already been developed by various leading companies, including PTC ThingWorx IIoT solution platform, Microsoft Azure IoT Hub, Hitachi Vantara IIoT platform, AWS IoT, Siemens MindSphere IIoT platform, and so forth.

Some preliminary industrial applications of IIoT have also been reported by researchers in academia. For example, Civerchia [28] developed an IIoT solution for the pervasive monitoring of industrial machinery through battery-powered IoT sensing devices. The proposed system was implemented in an electricity power plant where 33 temperature and vibration sensor devices were installed on different machinery to monitor their health status. This work shows a good example of using IIoT technology to support predictive maintenance of industrial systems. To achieve accurate supply-side energy modelling and energy consumption optimisation, Peng et al. [29] developed an IIoT-based data acquisition network to collect refined energy consumption information in an aluminium extrusions manufacturing system. The IIoT-based approach allows the acquired energy data to be correlated with job, machine, and process data, and hence creates production events that represent specific energy consumption processes. Salhaoui et al. [30] developed a smart IIoT monitoring and control system to achieve remote monitoring of a concrete batching plant using unmanned aerial vehicles (UAVs). The control of the UAVs was integrated with the plant's industrial control system through an IIoT gateway, which also sends the data collected by the UAVs to the cloud for advanced data analysis. Aiming to improve the resource efficiency of food manufacturing, Jagtap et al. [31] introduced an IIoT-based framework that uses various IoT-based hardware and software to achieve accurate real-time monitoring of food waste generation and use of energy and water. The developed prototype showed great advantages in analysing and optimising the food manufacturing processes compared to traditional paper-based systems.

The IIoT gateway is a crucial component in an IIoT system that collects the real-time data from field-level devices, performs data cleaning, formatting, processing, and transfers the data to the cloud. The specific forms and function requirements of the IIoT gateways depend on specific industrial needs. To improve data collection and communication efficiency, Chen et al. [32] proposed an IIoT gateway solution that combines a field-programmable gate array (FPGA)-based hardware bridge and multiple scalable microcontrollers. The proposed system was implemented in a machine tool spindle monitoring scenario where the spindle vibration and temperature data from multiple sensors were collected and processed in the IIoT gateway. EI Kaed et al. [33] developed a semantic rules engine for IIoT gateway to achieving dynamic and flexible rule-based monitoring and control of production facilities. The semantic rules engine allows the IIoT gateway to handle semantic queries and infer additional knowledge from previous experiences. To accelerate the optimisation process of high-level manufacturing planning, Leng et al. [34] proposed an IIoT system that uses Raspberry Pi-based IIoT gateways to allow the permissioned blockchain to interact with machines via smart contracts. The manufacturing events collected from machines were batched up and transferred via the IIoT gateways as transactions and recorded in the blockchain database. Nevertheless, IIoT gateways that are specifically designed for complex manufacturing equipment such as machine tools, 3D printers, and robots have not been widely studied. In fact, the data acquisition and communication requirements of manufacturing equipment are different from common sensors and actuators due to the complex data structure and communication interfaces of those manufacturing devices.

In the context of IIoT, a reliable database that can handle massive amounts of time-series data generated by industrial devices also becomes crucial. Recently, time-series database (TSDB) has been increasingly used in IIoT applications due to its better scalability and higher efficiency when dealing with time-series data compared with a relational database. Di Martino et al. [35] compared the performance of three TSDBs, including InfluxDB, Cassandra, and MongoDB, in the context of IIoT, and found that InfluxDB outperforms the other two for their specific time-series dataset. Costa et al. [36] developed an IIoT platform to support the data communication between the shop floor and multiple Industry 4.0 applications such as smart robotic additive manufacturing and adaptive pick and place robot. In their IIoT platform, InfluxDB was used as the TSDB to store the various types of real-time data collected from the robots, 3D printers, automated guided vehicles, and other sensors.

2.3 Research gaps

The literature review shows that previous research on cloud manufacturing mainly focused on the virtualisation of static manufacturing capability and high-level decision-making tasks. The link between the field-level manufacturing data (particularly the real-time data generated by manufacturing equipment) and the cloud manufacturing platform has not been well established. Recent advancements of IIoT have shown great potential in bridging this gap. However, IIoT gateways specifically designed for manufacturing equipment such as machine tools and 3D printers have not been widely studied. Besides, though existing resource virtualisation methods acknowledged that both the static capability information and dynamic availability information need to be captured, little research has been conducted on identifying the minimum amount of information about manufacturing equipment required for service selection and other decision-making tasks in cloud manufacturing. Research is required on building a generic service-oriented information model that captures the essential information for service management in cloud manufacturing.

To address these research gaps, we propose a service-oriented IIoT gateway solution in cloud manufacturing to link the field-level manufacturing equipment with the cloud manufacturing platform.

3. Service-oriented IIoT gateway for cloud manufacturing

This section introduces the proposed service-oriented IIoT gateway solution for cloud manufacturing. First, the system architecture of IIoT-supported cloud manufacturing is proposed. Second, the mechanism of the service-oriented IIoT gateway is explained. Third, the service-oriented data schemas for machine tools and 3D printers are proposed. Finally, the cloud TSDB and data transformation process are further elaborated.

3.1 System architecture of IIoT-supported cloud manufacturing

To illustrate how IIoT technologies can be applied to establish the link between the field-level manufacturing equipment and the cloud manufacturing platform in the context of cloud manufacturing, we propose a generic system architecture of IIoT-supported cloud manufacturing system as shown in Figure 1. This subsection explains the overall mechanism and workflow of the proposed cloud manufacturing system. Details of the core components on which this research focuses are further discussed in the following subsections.

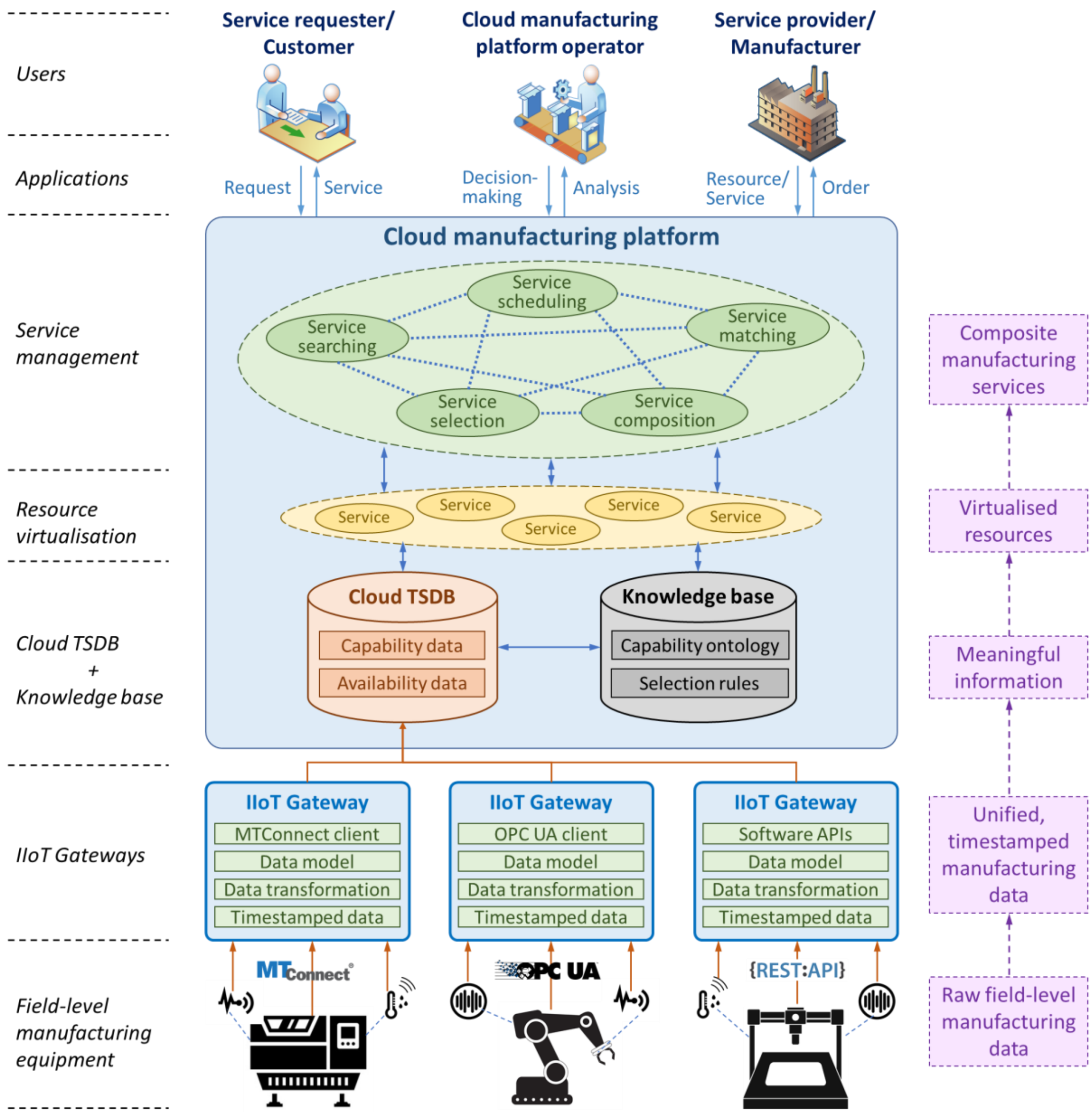

Figure 1. System architecture of IIoT-supported cloud manufacturing

The proposed IIoT-supported cloud manufacturing system follows the typical layered architecture of cloud manufacturing. The bottom layer represents the physical manufacturing resources provided by service providers or manufacturers. While manufacturing resources in the context of cloud manufacturing generally include all the hard resources (manufacturing facilities, computing facilities, materials, etc.) and soft resources (software, knowledge, personnel, etc.), the system architecture proposed in this research focuses specifically on the field-level manufacturing equipment such as machine tools, 3D printers, and robots. The manufacturing equipment is usually equipped with different types of sensors to collect additional real-time data of the manufacturing processes or the equipment's status.

To efficiently collect these heterogeneous manufacturing data and transfer them to the cloud manufacturing platform for resource virtualisation and further decision-making tasks, we proposed an additional layer of IIoT gateways to bridge the physical world and the cloud. In this layer, each IIoT gateway is connected to a piece of manufacturing equipment and its associated sensors. The IIoT gateway uses standardised communication protocols or software application programming interfaces (APIs) to acquire the real-time manufacturing data from the manufacturing equipment and sensors, assign a timestamp to each piece of the data, and transform them into a unified data format. The IIoT gateways transfer the unified, timestamped manufacturing data to a cloud time-series database (TSDB) in the cloud manufacturing platform through the Internet.

In the cloud manufacturing platform, the physical manufacturing equipment are virtualised as manufacturing services based on their feasibility to conduct certain manufacturing tasks. The feasibility of manufacturing equipment is mainly determined by two factors, i.e., capability and availability. The capability of manufacturing equipment is relatively static since its structure and functions do not usually change. On the other hand, the availability of manufacturing equipment is dynamically changing depending on their real-time status and job schedule. The data related to both capability and availability of the manufacturing equipment are transferred from the IIoT gateways to the cloud manufacturing platform and stored in the cloud TSDB. The cloud manufacturing platform also hosts a knowledge base specifically designed for the manufacturing equipment. The knowledge base contains capability ontology and selection rules of the manufacturing equipment that are defined and updated by the equipment providers. An example of the capability ontology and selection rules for a computer numerical control (CNC) machine tool has been demonstrated in our previous work [7].

Resource virtualisation can be achieved by combining the capability data and availability data in the cloud TSDB with the ontology and rules in the knowledge base. The physical manufacturing equipment is virtualised as various types of individual manufacturing services. Finally, service management tasks such as service searching, selection, composition, matching, and scheduling are performed by the platform operator to integrate the individual services provided by distributed manufacturing resources into composite manufacturing services that meet the service requesters' specific needs.

From users' perspective, the service requesters/customers only need to send their manufacturing service requests (product design files, quality requirements, delivery time, etc.) to the cloud manufacturing platform through applications and wait for the service to be delivered. The service providers/manufacturers need to not only provide their physical manufacturing equipment as manufacturing resources, but also ensure the field-level manufacturing data and the equipment-specific manufacturing knowledge can be efficiently transferred to and updated in the cloud manufacturing platform. The platform operators then only focus on the high-level decision-making tasks such as resource virtualisation and various service management tasks while assuming all the needed data and knowledge are already available in the cloud manufacturing platform.

Thus, the proposed IIoT-supported cloud manufacturing system enables an efficient data transformation process, as indicated on the right side of Figure 1. The raw field-level manufacturing data generated by manufacturing equipment is transformed into unified, timestamped data by the IIoT gateways. Then the data become meaningful information representing the capability and availability of the manufacturing equipment in the cloud manufacturing platform. Next, the manufacturing knowledge further transforms the meaningful information into virtualised resources in individual manufacturing services. Lastly, these services are integrated with other services and become composite manufacturing services that fulfil the customers' specific requirements.

3.2 Mechanism of the IIoT gateway

The proposed IIoT gateway is the focus of this research. In general, the IIoT gateway leverages the advantages of current IIoT technologies to achieve highly efficient data acquisition, transformation, communication, storage, query, analysis, and visualisation. In this research, the IIoT gateway is specifically designed as a critical component that enables the field-level manufacturing data to be transferred to and updated in the cloud manufacturing platform to support further resource virtualisation and service management. The IIoT gateway comprises four main modules: 1) import interface, 2) data schema, 3) data transformation module, and 4) export interface. The workflow of the proposed IIoT gateway in a cloud manufacturing system is depicted in Figure 2.

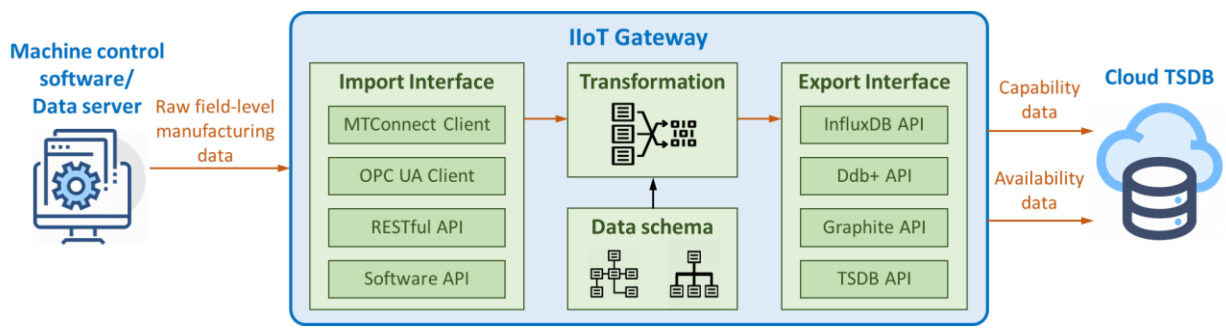

Figure 2. Workflow of the IIoT gateway in cloud manufacturing

Firstly, the raw field-level manufacturing data generated by manufacturing equipment and sensors are extracted from the machine control software or standardised data servers to the IIoT gateway through the import interface. Currently, more and more manufacturing equipment providers are attempting to make (part of) the data of their machines available to users by supporting open and standardised industrial communication protocols such as OPC UA [37] and MTConnect [38]. Despite their differences in data encoding, data modelling method, etc. [39], these two protocols both apply a similar server-client communication architecture and allow standardised real-time data communication. In this case, the MTConnect or OPC UA client can be implemented in the IIoT gateway as the import interface to extract the manufacturing data from the MTConnect agent or the OPC UA server of the manufacturing equipment. Alternatively, some manufacturing equipment uses open-source control software to develop their APIs (such as RESTful API) to extract data directly from the software. Secondly, a service-oriented data schema needs to be developed to organise all the collected manufacturing data. The data schema categorises the manufacturing data into capability data and availability data and defines the specific data items that need to be transferred to the cloud manufacturing platform. Details of the data schema will be further explained in Section 3.3. Thirdly, based on the data schema, the transformation module transforms the manufacturing data collected from the import interface into the format that can be accepted by the cloud TSDB. The specific target data format depends on the specific TSDB used in the cloud manufacturing platform. Details of the cloud TSDB and data transformation will be further explained in Section 3.4. Finally, the transformed capability data and availability data are streamed to the cloud TSDB in the cloud manufacturing platform through the export interface, which consists of the data input APIs of the corresponding cloud TSDB and an Internet connection to the cloud.

The proposed IIoT gateway has some distinct advantages in practical implementation compared to traditional field-level data acquisition systems. The IIoT gateway is a lightweight, low-cost solution that utilises various open-source software, APIs, communication protocols, and TSDBs. Common low-cost IoT-based microcontrollers such as Raspberry Pi and Arduino can be used as the hardware for the IIoT gateway. The standardised communication protocols, APIs, and ubiquitous Internet access endow the IIoT gateway with a PnP capability, such that different types of manufacturing equipment can be easily connected to the IIoT gateway without time-consuming and costly hardware configurations. The customisable data schema also provides the IIoT gateway with a service-oriented feature in line with the cloud manufacturing paradigm. In the data schema, the manufacturing equipment providers can define the specific data items they want to make available in the cloud manufacturing platform, thus providing only the data that best describe their services without disclosing other confidential manufacturing data. This can also reduce the network load between the IIoT gateways and the cloud.

3.3 Service-oriented data schemas for manufacturing equipment

Manufacturing equipment such as machine tools and 3D printers contain large volume and variety of data due to their complex structures and functions. In a cloud manufacturing platform, different data items of a manufacturing equipment will be used for different decision-making tasks such as service selection, composition, and scheduling. For ensuring these data can be efficiently queried and analysed in the cloud manufacturing platform, a service-oriented data schema for the manufacturing equipment needs to be developed. In the IIoT gateway, the data schema allows equipment providers to define and organize the specific data items they want to stream to the cloud. In the cloud manufacturing platform, the data schema is used as the reference to design the data structure of the cloud TSDB. Therefore, the data schema design for manufacturing equipment in a cloud manufacturing system needs to follow a service-oriented approach. On the one hand, the data schema should categorise the manufacturing data into different groups based on different services they are related to in the cloud manufacturing platform. On the other hand, the data schema should conform to the data structure requirement of the cloud TSDB and allow efficient data query and analysis for high-level decision-making tasks. Based on these requirements, this research proposes two generic service-oriented data schemas for machine tools and 3D printers, as shown in Figure 3 and Figure 4, respectively.

To allow easy implementation and data query for the cloud TSDB, the data schemas are designed as a tree structure with four levels. The first three levels represent the abstract data categories, and the fourth level represents the specific data items. As mentioned in the preceding sections, the feasibility of a manufacturing equipment to conduct certain services is determined mainly by two factors: capability and availability. The capability represents if the manufacturing equipment can conduct certain manufacturing tasks. In contrast, the availability represents if the manufacturing equipment is available to conduct specific manufacturing tasks at a particular time point. Therefore, the feasibility is firstly divided into capability data and availability data. The two data categories are then further divided into different sub-categories with each containing some specific data items.

The capability of manufacturing equipment is relatively static since it is determined by the equipment's intrinsic properties and functionalities that do not change over time unless its physical components are changed or performance degradation appeared after long time of use. In the context of cloud manufacturing, the equipment providers are responsible for providing accurate capability data of their manufacturing equipment. Most of the capability data (such as type of machine, table size, and max load) can be fetched from the manuals of the equipment, while some capability data (such as dimensional accuracy and max spindle speed) need to be provided based on the equipment providers' experience and updated periodically (e.g. yearly) to reflect the actual capability of the equipment due to long-term degradation.

The availability of manufacturing equipment is dynamically changing based on its real-time status. The availability data can be categorised into different levels such as production status, machine status, and process status. The higher level production status and machine status data can be directly used as dynamic availability information for resource virtualisation of the manufacturing equipment in the cloud manufacturing platform. The lower level process status data, on the other hand, can be further analysed with advanced data analytics tools to support various decision-making tasks such as service selection and composition in the cloud manufacturing platform. For example, the sensor data collected during machining processes such as motor current, vibration, and temperature can be used for pretrained machine learning models in the cloud to evaluate the machining quality such as surface roughness and machining accuracy. These evaluation results can be treated as QoS properties of the manufacturing service that can be used for service selection and composition.

The capability and availability information of a manufacturing resource collectively provides just-enough information for resource virtualisation, service composition, matching and scheduling. As discussed separately above, these two categories of information are the essential data inputs for manufacturing scheduling and optimisation in factories, though different types of resources may have slightly different data items under both categories.

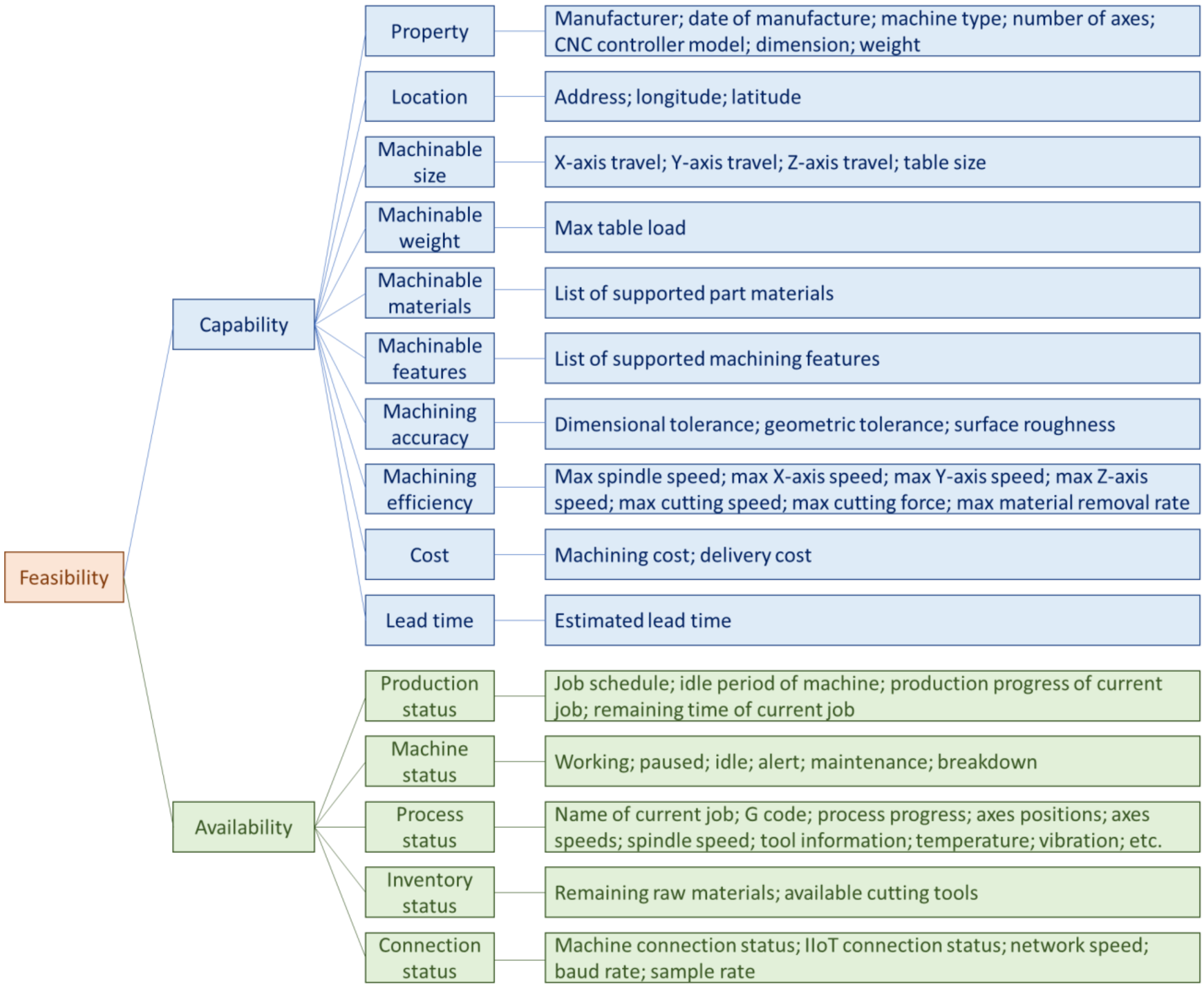

Figure 3. Service-oriented data schema for machine tools

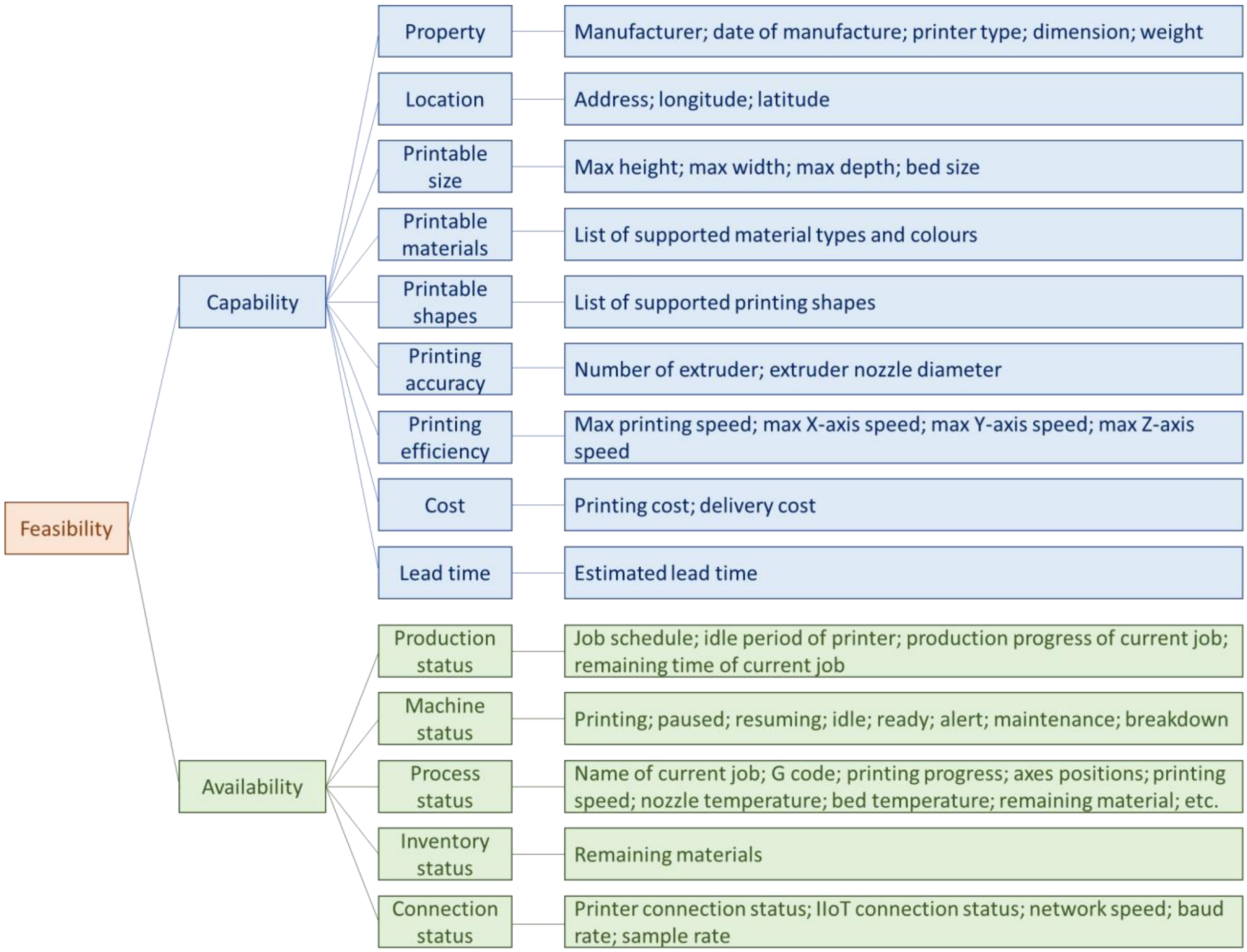

Figure 4. Service-oriented data schema for 3D printers

In the data schema for machine tools (Figure 3), the capability data comprises ten sub-categories. The property contains the data items representing the machine tool's basic information such as its manufacturer, type, controller model, and dimension. The location contains the specific location of the machine tool, which is essential for service selection. The machinable size, weight, materials, and features contain the part-related data items used in the capability ontology to evaluate if the machine tool can manufacture a specific part. The machining accuracy and efficiency contain the data items that determine if the machine tool can meet the quality and time requirements of the service requesters. The cost and time contain data items describing the cost and lead time of certain parts. The availability data comprises five sub-categories. The production status contains the real-time data related to the production progress and schedule assigned to the machine tool. The machine status represents the real-time working mode of the machine tool. The process status contains real-time machining data of the current process. Note that process status may contain confidential process parameters optional to be included by the equipment providers. The inventory status represents the available materials and cutting tools of the machine tool. The connection status contains data items that indicate the connectivity and network quality of the machine tool and its IIoT gateway.

The data schema for 3D printers (Figure 4) is similar to machine tools, though some sub-categories and specific data items are different due to the differences between 3D printers and machine tools. The detailed sub-categories and their corresponding commonly used data items for machine tools and 3D printers are listed in Figure 3 and Figure 4. It is worth mentioning that the proposed data schemas do not intend to provide an exhaustive list of all the data items for machine tools and 3D printers. While the four-level tree structure should be fixed to conform with the data structure in the cloud TSDB, the equipment providers could easily customise the data schema by editing the sub-categories and their corresponding data items. This allows the equipment providers to determine the specific manufacturing data they want to be exposed to the cloud manufacturing platform, thus fundamentally protecting their confidential data.

3.4 Cloud TSDB

The cloud TSDB is an essential component in the proposed IIoT-supported cloud manufacturing system. It resides in the cloud manufacturing platform and is directly linked with the field-level IIoT gateways through the Internet. All the field-level manufacturing data specified in the data schemas are streamed to the cloud TSDB from the connected IIoT gateways. The cloud TSDB stores those data in the cloud manufacturing platform and allows the platform operators to efficiently query, visualise, and analyse them to conduct high-level decision-making tasks.

As mentioned in the literature review, TSDB is chosen to be the cloud database because its advantages are scalability and efficiency compared to relational databases. TSDBs are specifically designed and optimised for time-series data, such as the data generated by various types of IIoT devices. TSDBs store data in chronological order and provide high concurrency and high throughput of data writing. Since both capability data and availability data need to be streamed to the cloud manufacturing platform from multiple IIoT gateways, TSDB becomes a good choice for the proposed cloud manufacturing system. Currently, there exist various open-source TSDBs that can be used to support the proposed low-cost IIoT gateway solution. Some popular open-source TSDBs include InfluxDB, Ddb+, Prometheus, Graphite, TimescaleDB, among others. Though the diversity of available open-source TSDBs provides more choices for the proposed cloud manufacturing system, it is noted that different TSDBs may apply different data structures in practical implementation. Meanwhile, the implementation of the proposed data schemas and the data transformation process both need to conform with a specific TSDB. Hence, to demonstrate the feasibility of the proposed approach, this research chooses one of the most popular and best-performing TSDBs, i.e., InfluxDB, to illustrate how the proposed data schemas can be used to develop the cloud TSDB for the IIoT gateways.

InfluxDB is an open-source TSDB that is purpose-built to handle the massive volume and multi-source time-series IoT data. It provides various built-in functions such as automatic time series transformation and aggregation that facilitate the efficient development of various applications for data query, visualisation, and analysis. Data communication with InfluxDB can be achieved through the prevalent network transport protocols such as HTTP (Hypertext Transfer Protocol), TCP (Transmission Control Protocol), and UDP (User Datagram Protocol). The database structure of InfluxDB is summarised in Figure 5. The top level of the database structure is the bucket that stores all InfluxDB data. Each bucket represents a database with an associated retention policy that defines the duration of time that each data point persists in that database. A bucket contains a set of measurements that are similar to the tables in relational databases. A measurement acts as a container for tags, fields, and timestamps. Tags and fields are both defined as key-value pairs. The tag key-value pairs are used to store the abstract metadata that can only be in string format. The field key-value pairs store the name and the actual value of a data item. In general, the combination of bucket name, measurement name, tag key-value pairs, and field keys can be used to query the values of specific data items stored in the TSDB.

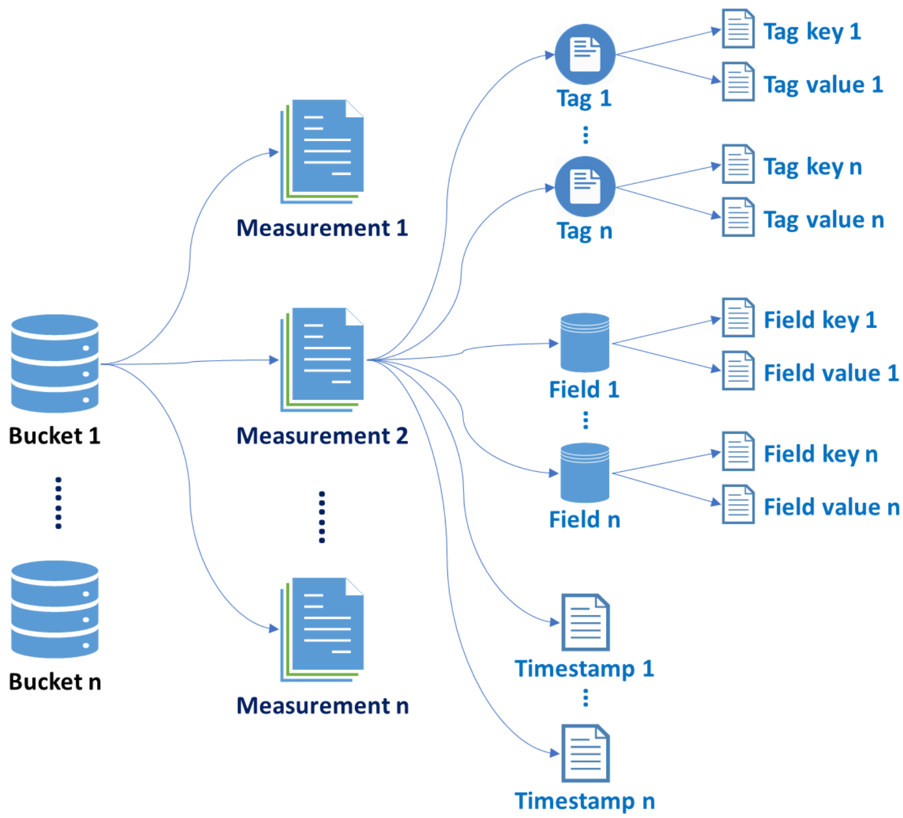

Figure 5. Database structure of InfluxDB

Since the service-oriented data schemas for manufacturing equipment introduced in the preceding subsection are designed as a four-level tree structure, they can be efficiently mapped to the database structure of InfluxDB. Specifically, the implementation of the proposed data schemas in InfluxDB is explained as follows. Firstly, since the capability data are relatively static, and the availability data are dynamically changing, different data retention policies should be applied to them in the TSDB. While the capability data should last in the TSDB for a long period as they are not frequently changed, the availability data that are continuously updated and accumulated in the TSDB should have a shorter retention period to keep the size of the TSDB within control. Hence, at the top level, capability data and availability data are separated into two individual buckets that have different retention policies. Secondly, each sub-category in the proposed data schemas (property, location, machine status, etc.) is mapped as a measurement in InfluxDB. These measurements represent the abstract groups of manufacturing data, and their names can be used for data query. Thirdly, the specific data items (manufacturer, address, spindle speed, etc.) and their values are mapped as the field key-value pairs in InfluxDB. Each field key-value pair is assigned with a timestamp. The field keys represent the data items' names, while the actual values are stored as the corresponding field values. In addition, a tag key-value pair is assigned to each measurement as the identification of the data source. The tag key represents the type of manufacturing equipment (machine tool, 3D printer, etc.), while the tag value represents the unique name or ID of the manufacturing equipment. In this way, the proposed schemas can be implemented as the database structure for the cloud TSDB and enable efficient data query.

3.5 Data transformation

As mentioned previously, in practical application, the manufacturing data collected from the control software and data servers need to be transformed in the IIoT gateways before streaming to the cloud TSDB. In this case, the data need to be transformed into the line protocol format as defined by InfluxDB. The line protocol format is a text-based format that contains four elements defined in the database structure, including measurement, tag key-value pairs, field key-value pairs, and timestamp. Figure 6 shows the typical arrangement of the elements in the line protocol format and a simplified example in practical application. Data are written into InfluxDB as multiple lines of text in the line protocol format in the data streaming process. Each line represents a data point that contains a measurement name, one or more tag key-value pairs, one or more field key-value pairs, and a timestamp.

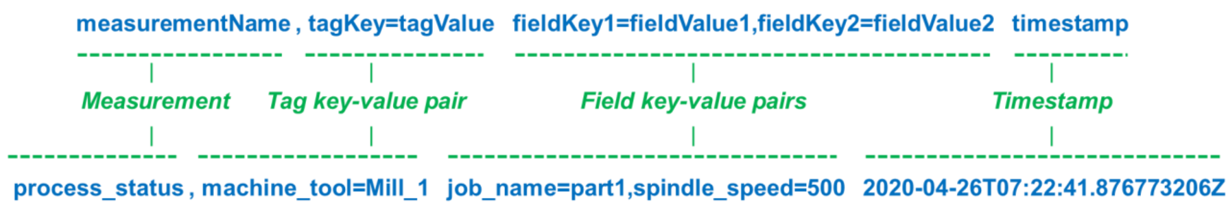

Figure 6. Elements and example of line protocol format in InfluxDB

Since the proposed data schemas have been mapped to the database structure, the aforementioned data transformation process can be achieved by encoding the sub-categories and data items in the data schema, the actual manufacturing data, and their timestamps as the corresponding elements in the line protocol. An example of the line protocol format data transformation of the process status data collected from a milling machine tool is demonstrated in the following pseudocode. In this example, the Point() method creates a data point and assigns process_status as the measurement name of the data point. The equipment type (machine_tool) and the ID of the equipment (Mill_1) are assigned to the data point as the tag key-value pair, respectively. The API.get() method represents a generic API that retrieves the raw manufacturing data from the control software or the data server. In this example, seven data items of process status have been collected, and their names and values are appended to the data point as field key-value pairs, respectively. The timestamp of the collected data is also retrieved through another generic API, API.datetime.now(), and appended to the data point. Finally, the pseudocode returns a transformed line of text containing all the retrieved process status data in the line protocol format that can be directly streamed to InfluxDB.

Pseudocode: Line protocol format data transformation

Input: Manufacturing data of process status collected from a milling machine tool
Output: Transformed line protocol format data

```
/* Create an empty line protocol */
    transformed_data = []
/* Create a data point and assign the sub-category as the measurement name */
    process_status = Point(“process_status”)
/* Assign the equipment type and equipment ID as a tag key-value pair to the data point */
    process_status.tag(“machine_tool”, Mill_1)
/* Append related data items and their values as field key-value pairs to the data point */
    process_status.string(“job_name”, API.get(“job_name”))
    process_status.string(“g_code”, API.get(“g_code”))
    process_status.double(“spindle_speed”, API.get(“spindle_speed”))
    process_status.double(“x_position”, API.get(“x_position”))
    process_status.double(“y_position”, API.get(“y_position”))
    process_status.double(“z_position”, API.get(“z_position”))
    process_status.double(“process_progress”, API.get(“process_progress”))
/* Assign a timestamp to the data point */
    process_status.datetime(API.datetime.now())
/* Return the data point as a line protocol */
    transformed_data.append(processStatusPoint)
```

4. Case studies

This section presents two IIoT gateways for a 3D printer and a CNC machine tool, respectively, to validate the feasibility and advantages of the proposed approach. Details of the field-level data acquisition and transformation, data schemas of the machines, data streaming and storage in the cloud TSDB, and data query and visualisation in the cloud are explained.

4.1 IIoT gateway for 3D printer

In this case study, a low-cost PnP IIoT gateway is developed for a 3D printer based on the proposed approach. The overall development process of this case study is briefly illustrated in Figure 7. The 3D printer used in this case is an Anycubic Kossel Linear Plus (Delta) 3D printer. A Raspberry Pi 3 Model B is used as the hardware for the IIoT gateway. InfluxDB is used as the cloud TSDB. The data query process is demonstrated by a data visualisation and analytics application developed with a dedicated open-source software named Grafana.

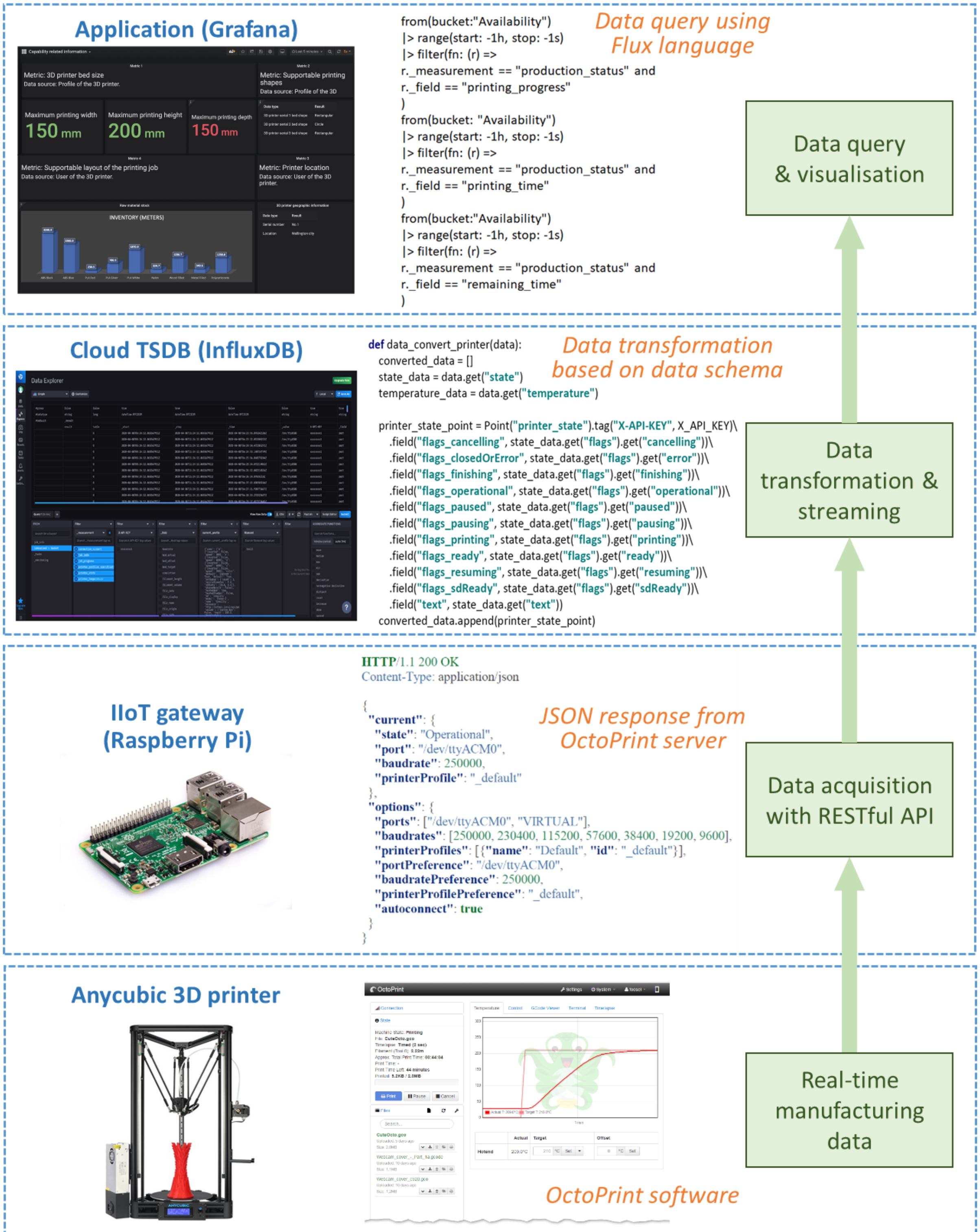

Figure 7. Development process of the IIoT gateway for 3D printer

The 3D printer is controlled by open-source control software, OctoPrint, which allows control and monitoring of the 3D printer using standard web browsers and is implemented on the Raspberry Pi. Since OctoPrint is open-source, real-time data acquisition of the 3D printer can be achieved by utilising self-developed APIs. In this case study, RESTful APIs are developed with an API development tool, Postman, to communicate with the OctoPrint server. The HTTP GET method is used to retrieve the specific data items from OctoPrint. In this case, since there is no additional sensor installed for this 3D printer, the availability of the real-time data items depends on the OctoPrint server. The real-time data collected from the OctoPrint server are originally represented in JSON format. Part of the JSON response retrieved by the RESTful API is demonstrated in Figure 7.

Upon analysing available real-time data that can be collected from the OctoPrint server, a concrete data schema for the Anycubic 3D printer is developed based on the generic data schema for 3D printers introduced in Section 3.3. Table 1 shows the detailed data schema corresponding to the InfluxDB data structure. The data source of each data item is also indicated. In this case, since the OctoPrint server only provides the real-time status data of the 3D printer, all the other relatively static data need to be manually inputted to the TSDB by the equipment provider.

Table 1. Data schema of the Anycubic 3D printer used in the case study

Bucket	**Measurement**	**Field key-value pair**	**Data source**
Capability	Property	Printer brand = Anycubic	Manual input
		Printer model = Kossel Linear Plus	Manual input
		Printer type = FDM	Manual input
		Dimension = 380*380*680 mm	Manual input
	Location	Location = Auckland city	Manual input
	Printable size	Max width = 230 mm	Manual input
		Max depth = 230 mm	Manual input
		Max height = 300 mm	Manual input
		Bed shape = Circle	Manual input
	Pintable materials	Printable materials = PLA, ABS, HIPS, Wood	Manual input
	Printing accuracy	Extruder quantity = 1	Manual input
		Nozzle diameter = 0.4 mm	Manual input
		X positioning accuracy = 0.0125 mm	Manual input
		Y positioning accuracy = 0.0125 mm	Manual input
		Z positioning accuracy = 0.0025 mm	Manual input
		Layer resolution = 0.1-0.4 mm	Manual input
	Printing efficiency	Max travel speed = 60 mm/s	Manual input
		Print speed = 20-60 mm/s	Manual input
Availability	Production status	Printing progress (%)	OctoPrint server
		Current printing time (min)	OctoPrint server
		Remaining printing time (min)	OctoPrint server
		Job schedule	Manual input
	Machine status	Printer state	OctoPrint server
	Process status	Print speed (mm/s)	OctoPrint server
		Layer height (mm)	OctoPrint server
		Nozzle temperature actual (°C)	OctoPrint server
		Nozzle temperature target (°C)	OctoPrint server
		Bed temperature actual (°C)	OctoPrint server
		Bed temperature target (°C)	OctoPrint server
	Inventory status	Material inventory	Manual input
	Connection status	Connection status	IIoT gateway
		Baud rate (KiB)	IIoT gateway

Based on the developed data schema, data transformation is performed in the IIoT gateway by encoding the manufacturing data of the 3D printer into line protocol format using python codes. On the one hand, the real-time JSON data retrieved from the OctoPrint server are parsed and converted into line protocol format following the pseudocode introduced in Section 3.4. On the other hand, the manual input data items are hardcoded directly into line protocol format texts. In the cloud TSDB, each bucket is assigned with a unique ID to allocate the related measurements. All the transformed data are streamed from the InfluxDB client in the IIoT gateway to their corresponding buckets in the cloud TSDB through the Internet. An authentication token is created and verified like a password during the data streaming process to protect data privacy and enhance data security for the equipment provider. Part of the data transformation code and the user interface of the cloud TSDB are demonstrated in Figure 7.

With the vast amount of field-level manufacturing data streamed to and stored in the cloud TSDB, efficient data query becomes a critical requirement for the subsequent resource virtualisation and service management tasks in the cloud manufacturing platform. Since the high-level decision-making tasks in the cloud manufacturing platform are not the focus of this research, a data visualisation application is developed in this case study to demonstrate the efficiency of data query and analytics in the proposed IIoT-supported cloud manufacturing system. The application is developed using Grafana Dashboard, an open-source web-based data query, analytics, and visualisation tool. The data query is performed using a standalone data scripting and query language named Flux. Flux allows efficient filtering and query of specific data items within a specific time period in the cloud TSDB based on the bucket ID, measurement name, and field key. Various data analytics functions, such as windowing, aggregating, and statistical calculations can also be performed during the query process. Part of the Flux query code used in this case study is demonstrated in Figure 7. In this case study, all the available data items in the cloud TSDB are queried using Flux codes, and the query results are imported to Grafana Dashboard for data visualisation. Some examples of the visualisation results are shown in Figure 8. The relatively static capability data are displayed as texts, while the changing availability data are displayed as different types of dynamically updated graphs. The data update rate can be customised in the Grafana Dashboard interface.

Overall, the field-level data of the 3D printer are continuously streamed from the IIoT gateway to the cloud TSDB, and simultaneously queried, updated, and visualised in the Grafana Dashboard. Hence, the developed application has validated the feasibility of the proposed approach. When the cloud TSDB is implemented in a cloud manufacturing platform, the platform operator can efficiently query the capability and availability data of all the connected field-level manufacturing equipment and perform resource virtualisation and various decision-making tasks in the cloud.

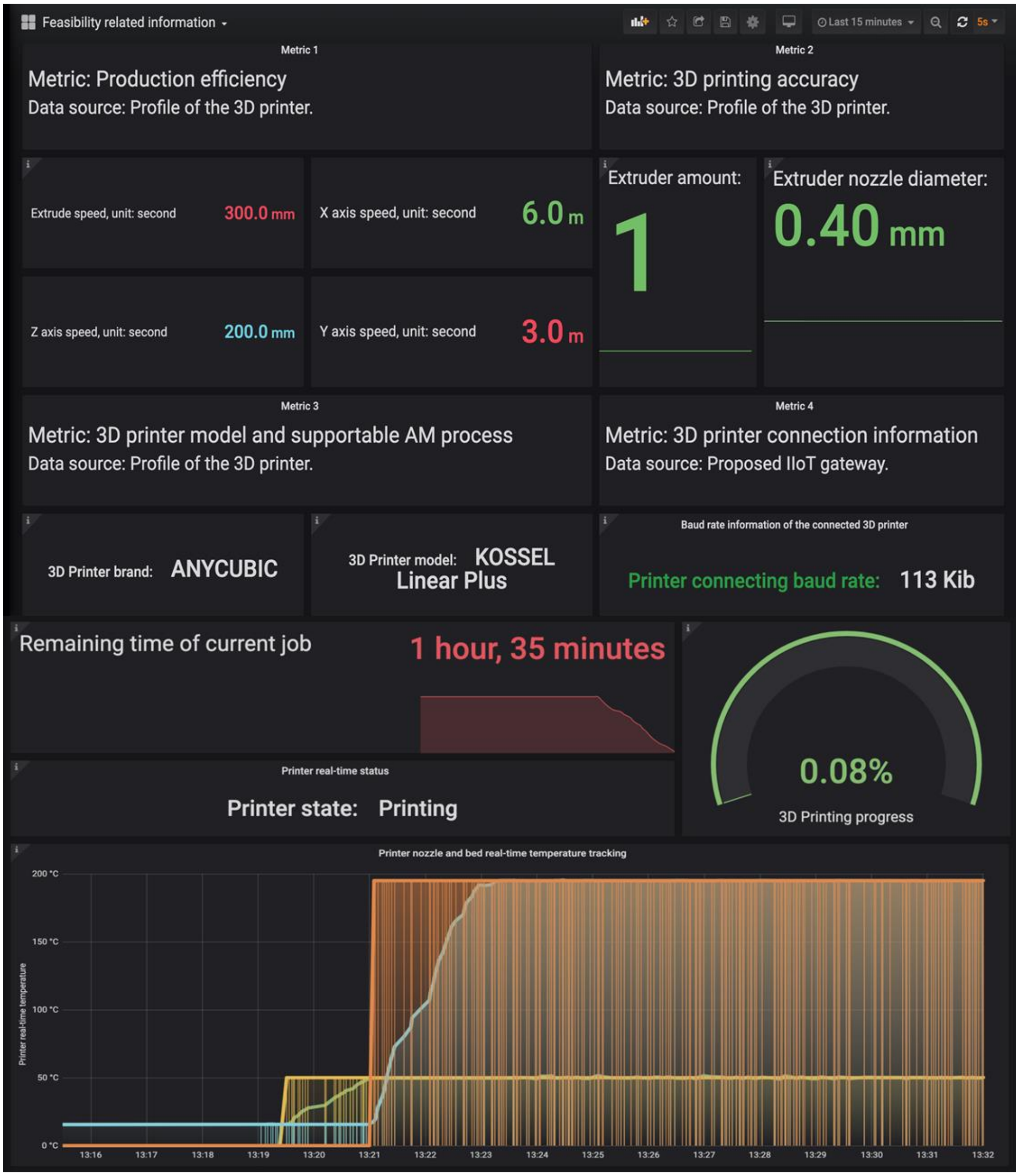

Figure 8. Data visualisation of query results using Grafana Dashboard

4.2 IIoT gateway for CNC machine tool

The second case study is the development of a low-cost PnP IIoT gateway for a CNC machine tool. The CNC machine tool used in this case is a 3-axis Sherline mill. The IIoT gateway is still developed using a Raspberry Pi. The general development procedures are the same as the previous case study and hence are not repeated here. Compared with the previous one, the major differences of this case study lie in the data acquisition and communication between the field level and the IIoT gateway. In this case, field-level manufacturing data acquisition from the machine tool is achieved based on our previous work on the MTConnect-enabled Cyber-Physical Machine Tool [40]. The overall system architecture of this case study is demonstrated in Figure 9.

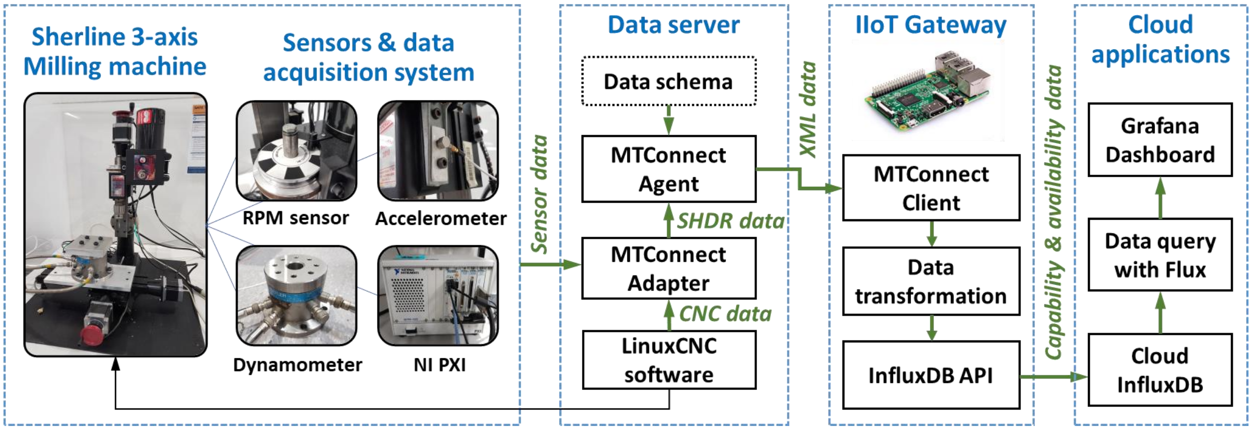

Figure 9. Overall system architecture of the IIoT gateway for CNC machine tool

At the field level, the CNC machine tool is controlled by an open-source control software named LinuxCNC. Different types of sensors and a data acquisition system are used to collect additional real-time process data from the machine tool. In the data server, an MTConnect Adapter is developed to retrieve the real-time machining data from both LinuxCNC software and external sensors. These raw data are converted into a unified data format (SHDR format) in the Adapter, assigned with timestamps, and then transmitted to the MTConnect Agent. The Agent organises all the field-level machining data based on a customised MTConnect data model and responds to data requests from MTConnect Clients. The MTConnect data model is essentially an XML file structured based on the rules defined by the MTConnect standard. It can contain both static values and real-time data items. Hence, in this case, the manual input data items in the data schema are stored and updated in the MTConnect data model by the equipment provider, such that no hard coding is needed in the data transformation process.

In this case study, the import interface of the IIoT gateway is developed as a simplified MTConnect Client, which communicates with the MTConnect Agent. The Client sends HTTP requests to the Agent and receives all the available machining data as an MTConnect-compliant XML file. These machining data, containing both capability and availability data, are well structured based on the data schema developed explicitly for the Sherline machine tool. Figure 10 demonstrates an example of part of the XML file that the IIoT gateway received from the data server. The example contains some real-time availability data of the machine tool, such as spindle speed, vibration, and controller status. Each data item is assigned with a unique ID to differentiate it from others. A timestamp is also attached to each data item to indicate the specific time point at which the data is collected. The black texts in Figure 10 represent the actual value of each data item.

```
▼<MTConnectStreams xmlns:m="urn:mtconnect.org:MTConnectStreams:1.3" xmlns="urn:mtconnect.org:MTConnectStreams:1.3"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.3
 http://www.mtconnect.org/schemas/MTConnectStreams_1.3.xsd">
   <Header creationTime="2020-12-19T05:37:26Z" sender="en-261175" instanceId="1513658141" version="1.3.0.17" bufferSize="131072"
   nextSequence="201485" firstSequence="70413" lastSequence="201484"/>
  ▼<Streams>
    ▼<DeviceStream name="Sherline-3Axis" uuid="000">
      ▼<ComponentStream component="Rotary" name="C" componentId="c1">
        ▼<Samples>
           <SpindleSpeed dataItemId="c2" timestamp="2020-12-19T05:37:25.979308" name="spindle_speed" sequence="201481"
           subType="ACTUAL">799.4775618715</SpindleSpeed>
           <Vibration dataItemId="c3" timestamp="2020-12-19T05:37:25.979308" name="vibMaxMag"
           sequence="201482">0.8349431792</Vibration>
         </Samples>
       </ComponentStream>
      ▼<ComponentStream component="Controller" name="controller" componentId="cn1">
        ▼<Events>
           <PowerState dataItemId="cn2" timestamp="2020-12-19T04:36:44.701327" name="power" sequence="3652">ON</PowerState>
           <EmergencyStop dataItemId="cn3" timestamp="2020-12-19T04:35:41.814842" name="alarm" sequence="28">ESTOP</EmergencyStop>
           <ControllerMode dataItemId="cn4" timestamp="2020-12-19T05:33:27.881859" name="mode"
           sequence="190915">MANUAL</ControllerMode>
           <Execution dataItemId="cn5" timestamp="2020-12-19T05:33:27.781685" name="execution" sequence="190905">READY</Execution>
         </Events>
       </ComponentStream>
```


Figure 10. XML format machining data received from the MTConnect agent

After receiving the field-level data from the data server, data transformation is then performed in the IIoT gateway by parsing the XML file, extracting the data items and their values based on the data schema, and converting the extracted data into line protocol format. The other tasks such as data query, analytics, and visualisation are achieved in the same way as introduced in the previous case study. Similar results have validated the feasibility of the proposed approach.

This case study has also demonstrated the distinct advantage of using standardised communication protocols in the proposed IIoT gateway system. Since MTConnect is an open-source, cross-platform, and standardised protocol, any CNC machine tool supporting MTConnect-based communication can be easily connected with the IIoT gateway in a PnP manner, despite their brands and models. The equipment providers only need to determine which data items they want to stream to the cloud manufacturing platform by customising the data model of the machine tool. Hence, the proposed low-cost PnP IIoT gateway solution can be easily implemented in large-scale manufacturing systems to support the development of the envisioned cloud manufacturing systems.

5. Conclusions and future work

Cloud manufacturing aims to allow ubiquitous and on-demand network access to a shared pool of various customisable manufacturing resources and services. While the high-level decision-making tasks in the cloud manufacturing platform, such as service selection, composition, matching, and scheduling, have been widely discussed and studied, the link between the field-level manufacturing data generated by various types of manufacturing equipment and the cloud manufacturing platform has not been well established.

In light of the great advantages shown by recent advancements in IIoT technologies, this paper proposes a service-oriented, low-cost, open-source, PnP IIoT gateway solution to realise efficient manufacturing data acquisition, communication, storage, query, and analysis in a cloud manufacturing system. The major contributions of this research work are summarised as follows:

- Proposed a generic system architecture of IIoT-supported cloud manufacturing system as a strategic guideline for integrating IIoT technologies in the cloud manufacturing environment.
- Proposed a service-oriented PnP IIoT gateway solution for efficient data acquisition, communication, query, analysis, and visualisation of manufacturing equipment in a cloud manufacturing system.
- Developed two generic service-oriented data schemas for machine tools and 3D printers that allow efficient data management and query for the decision-making tasks in the cloud manufacturing platform.
- Developed two practical PnP IIoT gateways for a 3D printer and a CNC machine tool that validate the feasibility of the proposed approach.

The proposed IIoT-supported cloud manufacturing system enables huge amounts and various types of field-level manufacturing data to be efficiently collected and transmitted to the cloud manufacturing platform, thus establishing a link between field-level manufacturing processes and cloud-based decision-making activities. The utilisation of IIoT techniques in cloud manufacturing significantly enhances the capability of resource virtualisation of various shop floor manufacturing equipment. The developed service-oriented data schemas allow both capability data and availability data of the manufacturing equipment to be captured and managed efficiently, thus facilitating the following decision-making tasks such as service selection and composition in the cloud manufacturing platform. Furthermore, the customisable data schemas allow the equipment providers to determine the specific data items they want to stream to the cloud, and hence protect their data privacy fundamentally. Based on the data schemas, a cloud TSDB is developed to enable efficient data storage, query, visualisation, and analysis in the cloud manufacturing platform. To demonstrate the feasibility and advantages of the proposed approach, we have developed two practical PnP IIoT gateways for a 3D printer and a CNC machine tool, respectively. The developed IIoT gateways demonstrated distinct advantages over traditional data acquisition systems due to the extensive utilisation of various low-cost hardware, open-source APIs and software solutions, and standardised communication protocols. Experimental results proved that the proposed IIoT gateway solution has a great PnP capability, and hence can be easily implemented in real-world, large-scale, complex manufacturing systems. Therefore, the proposed approach represents a low-cost, flexible, and efficient solution to practically transforming traditional manufacturing systems into cloud manufacturing systems.

The concept of cloud manufacturing has been introduced for a decade and has attracted enormous attention from both academia and industry. Though the great advantages of cloud manufacturing have been commonly recognised and a vast amount of research has been conducted, practical development of the envisioned cloud manufacturing systems is still at the preliminary stage. Efficient data acquisition, communication, storage, query, and analysis of the field-level manufacturing equipment have been a significant challenge that hinders the development of cloud manufacturing systems. This research investigates the implementation of IIoT technologies in a cloud manufacturing system to address this challenge. Results from this research indicate that deep integration of various emerging IIoT technologies in manufacturing systems could establish the link between the field-level manufacturing equipment and the cloud manufacturing platform, and hence significantly accelerate the realisation of the envisioned cloud manufacturing paradigm.

Based on this research work and our previous research on resource virtualisation [7,10], service composition [18], and big data analytics [9] in cloud manufacturing, our future work will focus on dynamic manufacturing resource virtualisation in the IIoT-supported cloud manufacturing environment. The knowledge base (as mentioned in Figure 1) that contains both capability ontology and availability selection rules of the manufacturing equipment will be developed. Integration of the field-level manufacturing data and the knowledge base will be studied to realise dynamic resource virtualisation of different manufacturing equipment in the cloud manufacturing platform. Furthermore, various types of service management tasks based on the proposed cloud manufacturing system could also be investigated.

Declaration of Competing Interest

The authors reported no potential conflict of interest.

References

[1] Xu X. From cloud computing to cloud manufacturing. Robotics and Computer-Integrated Manufacturing 2012;28:75–86. https://doi.org/10.1016/j.rcim.2011.07.002.

[2] He W, Xu L. A state-of-the-art survey of cloud manufacturing. International Journal of Computer Integrated Manufacturing 2015. https://doi.org/10.1080/0951192X.2013.874595.

[3] Zhang L, Luo Y, Tao F, Li BH, Ren L, Zhang X, et al. Cloud manufacturing: a new manufacturing paradigm. Enterprise Information Systems 2014;8:167–87. https://doi.org/10.1080/17517575.2012.683812.

[4] Li B-H, Zhang L, Wang S-L, Tao, Fei and Cao J, Jiang X, Song X, et al. Cloud manufacturing: a new service-oriented networked manufacturing model. Computer Integrated Manufacturing Systems 2010. https://doi.org/1.

[5] Ghomi EJ, Rahmani AM, Qader NN. Cloud manufacturing: challenges, recent advances, open research issues, and future trends. International Journal of Advanced Manufacturing Technology 2019. https://doi.org/10.1007/s00170-019-03398-7.

[6] Borangiu T, Trentesaux D, Thomas A, Leitão P, Barata J. Digital transformation of manufacturing through cloud services and resource virtualization. Computers in Industry 2019. https://doi.org/10.1016/j.compind.2019.01.006.

[7] Lu Y, Xu X. Resource virtualization: A core technology for developing cyber-physical production systems. Journal of Manufacturing Systems 2018;47:128–40. https://doi.org/10.1016/j.jmsy.2018.05.003.

[8] Sisinni E, Saifullah A, Han S, Jennehag U, Gidlund M. Industrial internet of things: Challenges, opportunities, and directions. IEEE Transactions on Industrial Informatics 2018;14:4724–34. https://doi.org/10.1109/TII.2018.2852491.

[9] Lu Y, Xu X. Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services. Robotics and Computer-Integrated Manufacturing

2019;57:92–102. https://doi.org/10.1016/j.rcim.2018.11.006.

[10] Lu Y, Shao Q, Singh C, Xu X, Ye X. Ontology for manufacturing resources in a cloud environment. International Journal of Manufacturing Research 2014;9:448–69. https://doi.org/10.1504/IJMR.2014.066666.

[11] Liu N, Li X, Shen W. Multi-granularity resource virtualization and sharing strategies in cloud manufacturing. Journal of Network and Computer Applications 2014;46:72–82.

[12] Luo Y, Zhang L, Tao F, Ren L, Liu Y, Zhang Z. A modeling and description method of multidimensional information for manufacturing capability in cloud manufacturing system. International Journal of Advanced Manufacturing Technology 2013. https://doi.org/10.1007/s00170-013-5076-9.

[13] Tao F, Cheng Y, Xu L Da, Zhang L, Li BH. CCIoT-CMfg: Cloud computing and internet of things-based cloud manufacturing service system. IEEE Transactions on Industrial Informatics 2014;10:1435–42. https://doi.org/10.1109/TII.2014.2306383.

[14] Ren L, Zhang L, Zhang YB, Tao F, Luo YL. Resource virtualization in cloud manufacturing. Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS 2011;17:511–8. https://doi.org/n.

[15] Tao F, Laili Y, Xu L, Zhang L. FC-PACO-RM: A parallel method for service composition optimal-selection in cloud manufacturing system. IEEE Transactions on Industrial Informatics 2013. https://doi.org/10.1109/TII.2012.2232936.

[16] Lartigau J, Xu X, Nie L, Zhan D. Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm. International Journal of Production Research 2015:1–25.

[17] Yuan M, Zhou Z, Cai X, Sun C, Gu W. Service composition model and method in cloud manufacturing. Robotics and Computer-Integrated Manufacturing 2020. https://doi.org/10.1016/j.rcim.2019.101840.

[18] Lu Y, Xu X. A semantic web-based framework for service composition in a cloud manufacturing environment. Journal of Manufacturing Systems 2017;42:69–81. https://doi.org/10.1016/j.jmsy.2016.11.004.

[19] Yang B, Wang S, Li S, Jin T. A robust service composition and optimal selection method for cloud manufacturing. International Journal of Production Research 2020. https://doi.org/10.1080/00207543.2020.1852481.

[20] Liang H, Wen X, Liu Y, Zhang H, Zhang L, Wang L. Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning. Robotics and Computer-Integrated Manufacturing 2020;67. https://doi.org/10.1016/j.rcim.2020.101991.

[21] Yu C, Zhang L, Zhao W, Zhang S. A blockchain-based service composition architecture in cloud manufacturing. International Journal of Computer Integrated Manufacturing 2020. https://doi.org/10.1080/0951192X.2019.1571234.

[22] Liu Y, Wang L, Wang XV, Xu X, Zhang L. Scheduling in cloud manufacturing: state-of-the-art and research challenges. International Journal of Production Research 2019. https://doi.org/10.1080/00207543.2018.1449978.

[23] Li W, Zhu C, Yang LT, Shu L, Ngai ECH, Ma Y. Subtask Scheduling for Distributed Robots in Cloud Manufacturing. IEEE Systems Journal 2017. https://doi.org/10.1109/JSYST.2015.2438054.

[24] Liu Y, Xu X, Zhang L, Wang L, Zhong RY. Workload-based multi-task scheduling in cloud manufacturing. Robotics and Computer-Integrated Manufacturing 2017. https://doi.org/10.1016/j.rcim.2016.09.008.

[25] Wang LC, Chen CC, Liu JL, Chu PC. Framework and deployment of a cloud-based advanced planning and scheduling system. Robotics and Computer-Integrated Manufacturing 2021;70. https://doi.org/10.1016/j.rcim.2020.102088.

[26] Boyes H, Hallaq B, Cunningham J, Watson T. The industrial internet of things (IIoT): An analysis framework. Computers in Industry 2018. https://doi.org/10.1016/j.compind.2018.04.015.

[27] Gartner Magic Quadrant for Industrial IoT Platforms n.d. https://www.gartner.com/en/documents/3991952/magic-quadrant-for-industrial-iot-platforms (accessed June 5, 2021).

[28] Civerchia F, Bocchino S, Salvadori C, Rossi E, Maggiani L, Petracca M. Industrial Internet of Things monitoring solution for advanced predictive maintenance applications. Journal of Industrial Information Integration 2017. https://doi.org/10.1016/j.jii.2017.02.003.

[29] Peng C, Peng T, Liu Y, Geissdoerfer M, Evans S, Tang R. Industrial Internet of Things enabled supply-side energy modelling for refined energy management in aluminium extrusions manufacturing. Journal of Cleaner Production 2021;301. https://doi.org/10.1016/j.jclepro.2021.126882.

[30] Salhaoui M, Guerrero-González A, Arioua M, Ortiz FJ, El Oualkadi A, Torregrosa CL. Smart industrial iot monitoring and control system based on UAV and cloud computing applied to a concrete plant. Sensors (Switzerland) 2019. https://doi.org/10.3390/s19153316.

[31] Jagtap S, Garcia-Garcia G, Rahimifard S. Optimisation of the resource efficiency of food manufacturing via the Internet of Things. Computers in Industry 2021;127. https://doi.org/10.1016/j.compind.2021.103397.

[32] Chen CH, Lin MY, Liu CC. Edge Computing Gateway of the Industrial Internet of Things Using Multiple Collaborative Microcontrollers. IEEE Network 2018;32:24–32. https://doi.org/10.1109/MNET.2018.1700146.

[33] Kaed C El, Khan I, Van Den Berg A, Hossayni H, Saint-Marcel C. SRE: Semantic rules engine for the industrial internet-of-things gateways. IEEE Transactions on Industrial Informatics 2018. https://doi.org/10.1109/TII.2017.2769001.

[34] Leng J, Yan D, Liu Q, Xu K, Zhao JL, Shi R, et al. ManuChain: Combining Permissioned Blockchain with a Holistic Optimization Model as Bi-Level Intelligence for Smart Manufacturing. IEEE Transactions on Systems, Man, and Cybernetics: Systems 2020. https://doi.org/10.1109/TSMC.2019.2930418.

[35] Di Martino S, Fiadone L, Peron A, Vitale VN, Riccabone A. Industrial Internet of Things: Persistence for Time Series with NoSQL Databases. Proc. - 2019 IEEE 28th Int. Conf. Enabling Technol. Infrastruct. Collab. Enterp. WETICE 2019, 2019.

https://doi.org/10.1109/WETICE.2019.00076.

[36] Costa FS, Nassar SM, Gusmeroli S, Schultz R, Conceição AGS, Xavier M, et al. Fasten iiot: An open real-time platform for vertical, horizontal and end-to-end integration. Sensors (Switzerland) 2020. https://doi.org/10.3390/s20195499.

[37] OPC Foundation. OPC UA (Unified Architecture) 2021.

[38] MTConnect Institute. MTConnect standard 2021.

[39] Liu C, Hong X, Zhu Z, Xu X. Machine Tool Digital Twin: Modelling methodology and applications. Proc. Int. Conf. Comput. Ind. Eng. CIE, 2018.

[40] Liu C, Vengayil H, Zhong RY, Xu X. A systematic development method for cyber-physical machine tools. Journal of Manufacturing Systems 2018;48:13–24. https://doi.org/10.1016/j.jmsy.2018.02.001.