Journal of Manufacturing Systems 55 (2020) 348–359

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Semantic communications between distributed cyber-physical systems towards collaborative automation for smart manufacturing

Yuqian Lu[a],*, Muhammad Rizwan Asghar[b]

[a] Department of Mechanical Engineering, The University of Auckland, New Zealand
[b] School of Computer Science, The University of Auckland, New Zealand

ARTICLE INFO

Keywords:
Machine-to-machine communication
Cyber-physical system
Smart manufacturing
Semantic web
Industrial Internet of Things (IIoT)
Semantic-aware cyber-physical system
Knowledge graph

ABSTRACT

Machine-to-machine (M2M) communication is a crucial technology for collaborative manufacturing automation in the Industrial Internet of Things (IIoT)-empowered industrial networks. The new decentralized manufacturing automation paradigm features ubiquitous communication and interoperable interactions between machines. However, peer-to-peer (P2P) interoperable communications at the semantic level between industrial machines is a challenge. To address this challenge, we introduce a concept of Semantic-aware Cyber-Physical Systems (SCPSs) based on which manufacturing devices can establish semantic M2M communications. In this work, we propose a generic system architecture of SCPS and its enabling technologies. Our proposed system architecture adds a semantic layer and a communication layer to the conventional cyber-physical system (CPS) in order to maximize compatibility with the diverse CPS implementation architecture. With Semantic Web technologies as the backbone of the semantic layer, SCPSs can exchange semantic messages with maximum interoperability following the same understanding of the manufacturing context. A pilot implementation of the presented work is illustrated with a proof-of-concept case study between two semantic-aware cyber-physical machine tools. The semantic communication provided by the SCPS architecture makes ubiquitous M2M communication in a network of manufacturing devices environment possible, laying the foundation for collaborative manufacturing automation for achieving smart manufacturing. Another case study focusing on decentralized production control between machines in a workshop also proved the merits of semantic-aware M2M communication technologies.

1. Introduction

With the pervasive use of emerging Information and Communication Technologies (ICT), the manufacturing industry is entering a new era of fully-integrated autonomous environment, i.e., smart manufacturing, that can dynamically respond to changes in system status, customer needs and supply chain network [1,2]. Smart manufacturing systems will work collaboratively with minimum human involvement towards optimized operations for great production flexibility and product variability [3,4]. Smart manufacturing relies on M2M communication that enables smooth data exchange and interaction between manufacturing things (i.e., devices, machines, systems, and humans), which then allows dynamic configuration and autonomous collaboration between them. However, communication, interoperability, and collaboration between connected machines are still challenging tasks [5]. Although the crucial technical issues of network connectivity [6] have been addressed adequately, the technologies are not ready for communication between heterogeneous machines in a flexible and seamless manner [5]. Ubiquitous M2M messaging and machine understanding are the central issues of autonomous machine interactions in the context of smart manufacturing. In particular, the successful development of smart manufacturing ultimately relies on interoperable communications at the semantic level [7].

In this work, we developed systematic technologies for enabling semantic M2M communications between connected manufacturing things, which can be easily extended to different application contexts. The focus of the research is to explore the methods of establishing semantic-rich communications between manufacturing systems, i.e., requiring an investigation of upgrading conventional Cyber-Physical Systems (CPSs) to SCPSs. The developed technologies can enable the creation of semantic awareness between manufacturing systems/devices based on which further cognitive capabilities can be developed to enable autonomous collaboration between manufacturing systems. Although this research mainly focuses on CPS, the methodologies and technologies can directly apply to manufacturing things in general.

In summary, we make the following research contributions in this

---

* Corresponding author.
E-mail addresses: yuqian.lu@auckland.ac.nz (Y. Lu), r.asghar@auckland.ac.nz (M.R. Asghar).

https://doi.org/10.1016/j.jmsy.2020.05.001
Received 17 October 2019; Received in revised form 2 May 2020; Accepted 3 May 2020

work.

- We proposed the concept of semantic-aware CPSs that can enable semantic M2M communications between CPSs in the context of collaborative smart manufacturing automation.
- We presented a generic layered architecture for enabling semantic-aware CPSs via developing a communication layer and semantic layer on top of the existing CPS system architectures. The separation between CPS internal architecture and its external communication concerns ensures a smooth upgrade of a CPS to an SCPS.
- We demonstrated the practical implementation of the proposed SCPS concept with state-of-the-art semantic enrichment technologies.

The rest of the paper is as follows. Section 2 reviews existing practices of M2M communications and outlines the research gaps for developing semantic M2M communications in a heterogeneous manufacturing environment. Section 3 explains the concept of semantic interactions between connected manufacturing systems and proposes a generic architecture of semantic-aware CPSs that is compatible with the Cyber-Physical Production Systems (CPPSs) [4,8] and Asset Administration Shell (AAS) framework [9]. Section 4 presents the implementation of the proposed architecture and a case study on enabling semantic M2M communications between two machine tools with the developed technologies. An illustrative test of the developed technologies in a smart manufacturing workshop for distributed production process control is also discussed in Section 4. Sections 5 and 6 discuss various aspects of this work, conclude the paper, and point out future research directions.

2. State-of-the-art analysis

Future manufacturing is characterized by the autonomous configuration and collaboration between connected devices (in the form of CPSs) that can make optimal decisions with minimum human involvement [10]. To this end, ubiquitous communication and understanding between connected machines become a crucial technological foundation. In order to communicate, manufacturing devices must be able to:

- Send and receive messages – at this **physical level**, manufacturing devices must communicate over agreed physical and network layers to be able to send and receive objects that represent messages;
- Parse the messages – at the **syntactic level**, manufacturing devices must be able to parse messages to correctly decode the message to its parts, such as message content, language, sender, and must be able to parse the content of the message;
- Understand the messages – at the **semantic level**, manufacturing devices must interpret and reason about the parsed metadata in the same context in the same way, in which context-aware actions can be taken.

Intelligent communications and autonomous decision-making between collaborative manufacturing devices require communications to be established at the semantic level [11]. This section reviews the prior research efforts on M2M communications in the manufacturing domain and identifies the research gaps for achieving semantic communications between machines to satisfy collaborative automation in smart manufacturing.

2.1. M2M communication channels – physical level technologies

M2M communication refers to direct communication between devices using a communication channel, which can be wired or wireless [6]. The foundation of M2M communications is the reliable exchange of information. To facilitate information exchange, a multitude of industrial communication networks evolved over the years, starting from the 1980s. According to [6], industrial communications started with dedicated fieldbus networks, such as PROFIBUS and Modbus to enable M2M communications. However, many fieldbus protocols were designed to operate on different physical media and have wide compatibility issues with the OSI model [12]. Driven by the advancement of Internet technologies, Ethernet-based networks became popular for facilitating inter-communication at a higher level. Since 2000, the concept of Internet of Things (IoT) and Wireless Sensor Networks (WSN) have been impacting the industrial network field. Some modern approaches are adopting existing standards such as IEEE 802.11, IEEE 802.15.1, and IEEE 802.15.4. Detailed discussion on these transport-oriented network communication channels and their newer development is presented in [6].

2.2. M2M communication information models – syntactic level technologies

The above transport-oriented communication technologies like fieldbuses, industrial Ethernet and industrial wireless approaches enable industrial data exchange with guaranteed reliability, availability and real-time behavior. However, these communication technologies are not enough for achieving meaningful data exchange in a specific domain. A meta-model that explicitly defines the data content to be exchanged is necessary [13].

Focusing on industrial automation data communication, OPC Foundation developed OPC Unified Architecture (OPC UA) to provide a cross-platform M2M communication mechanism for exchanging and integrating data between industrial devices and systems [14]. Furthermore, OPC UA supports an integral object-oriented information modeling approach to allow the development of vendor-specific companion models, which has attracted much research efforts in developing domain-specific information models. For example, Miyazawa et al. translated IEC 61131-3 software model to an OPC UA compatible information model, recognizing that OPC UA provides a suitable data exchange mechanism [15]. Trnka et al. developed an OPC UA information model for large-scale process control applications [16]. Maka et al. presented an OPC UA information model for real-time management of a large amount of data coming from public transportation system [17]. In the manufacturing domain, MTConnect consortium developed a semantic vocabulary for manufacturing equipment to provide structured, contextualized data without any proprietary format [18]. This standard has been widely used for digitizing manufacturing activities, such as for monitoring machining processes [19] and machining simulation [20]. In particular, Liu et al. developed an interoperable cyber-physical machine tool platform based on OPC UA and MTConnect [21].

However, the multitude of information modeling methods and developed specifications create data silos that hinder wide scope data exchange. One solution to solve this problem is to develop data conversion tools, such as conversion between OPC UA and UML Model[22], OPC UA and MTConnect [23], and OPC UA and AutomationML [24]. These methods though to some extent solved the issue of cross-application communication; they still lack the required semantic understandings for intelligent decisions between manufacturing devices and systems.

2.3. M2M communication semantic interoperability – semantic level technologies

Semantic level technologies are responsible for interpreting message content and building contextual understandings in a domain. Targeting at building semantic understandings between two communication bodies and the wider domain, researchers have tried methods to add semantics to data communication. For example, Hrvoje et al. introduced extensions of the Rich Presence Information Data (RPID) as a tool to address context information in M2M communications [25]. To enable collaboration between machines, Meng et al. [5] proposed a collaboration-oriented M2M messaging mechanism. These research works extended the use of data communications between machines by further analyzing data involving a context. However, existing research lacks the autonomy of reasoning at the semantic level, which restricts the flexibility of distributed intelligence between machines in a decentralized collaboration environment.

Aiming at developing semantic communications between manufacturing systems and even in an IIoT environment, Grangel-González et al. added semantic interoperability to Industry 4.0 components using semantic technologies [26]. This work extended the Asset Administrative Shell concept by representing the digital profile based on semantic knowledge representation formalisms, such as RDF, RDF Schema and OWL. With the same intention, a follow-up piece of work demonstrated the feasibility and benefits of using RDF to model Industry 4.0 components, as well as demonstrated the possibility of integrating with other related standards in the manufacturing domain [27]. Willner et al. validated the concept of semantic communications between distributed systems based on semantic-aware oneM2M specification [7]. This research utilized Semantic Web technologies to map oneM2M constructs to ontologies and increased interoperability between two remote virtual machines. Similarly, Kirkizh et al. presented a conceptual mechanism of enabling interoperable communications between hardware and software applications belonging to different domains using Semantic Web technologies [28].

2.4. Open issues

Future collaborative manufacturing automation demands flexible, reconfigurable, scalable, and interoperable network-enabled collaboration between decentralized and distributed CPSs to satisfy the need for rapid production of personalized products with maximized business agility. Manufacturing systems will become intelligent that rely on real-time P2P communication and reasoning to make informed decisions to respond to dynamic changes inside and outside the systems. In particular, achieving semantic communication and understanding between distributed CPSs in a network is of paramount importance for making collaborative manufacturing automation a reality [29].

Though existing standards, e.g., MTConnect, OPC-UA, and AutomationML, allow for specifications of industrial objects and information-rich M2M communications, the information models generated from these standards are not semantically defined, making the semantic understanding and intelligent decision-making a challenge [30]. The relevant studies do not achieve integral interoperability at the connectivity, communication, and semantic understanding as a whole to accommodate the specificities of industrial scenarios and make it well suited for smart manufacturing applications. In summary, a technical gap restricts the data and information level technologies from truly integrating with machine interactions and industrial operations to leverage machine interoperability and high-level adaptability. Research needs to focus on developing robust semantic M2M communications for achieving collaborative automation between complex machines and manufacturing systems via integrating the existing physical communication protocols, information modeling and semantic enrichment technologies. A generic system architecture of semantic-aware manufacturing systems is urgently required for building the foundation of scalable M2M communications between manufacturing systems, regardless of the heterogeneous physical models and operation models of manufacturing systems in reality. Aiming at advancing semantic M2M communications, we report a generic architecture of semantic-aware manufacturing systems and the enabling technologies, with a focus on integrating with existing CPS architectures, such as 5C architecture [31] and AAS [32], and standard information model for manufacturing things, such as MTConnect vocabulary [33].

3. Semantic-aware cyber-physical systems

This section presents a high-level system architecture for SCPS to enable secure semantic M2M communications in the context of autonomous collaborations between connected CPSs. This generic system architecture allows manufacturers to develop flexible P2P M2M communications and use this flexible semantic communication mechanism to build collaborative manufacturing automation systems. Several specific aspects of the SCPS system architecture, including its decoupled layered architecture and implementation principles, are presented in the following subsections.

3.1. System architecture

Conventional CPSs were developed for managing systems between its physical assets and computational capabilities [31], with little attention to the communication and interactions between CPSs in a networked environment. However, distributed collaborative systems are the future norms of Industry 4.0 [3]. The proposed architecture was designed to be generic, flexible and extendable, and can be easily integrated with existing CPSs. Acknowledging the heterogeneity in CPS architecture, we propose additional layers on a CPS to ensure a CPS, regardless of its internal architecture, can be easily upgraded to an SCPS.

Fig. 1 depicts a conceptual model that standardizes the communication mechanisms between two Semantic-aware CPSs. Its goal is to establish interoperable semantic communications between CPSs with standard communication protocols. The model partitions an SCPS into four abstraction layers, i.e., a Physical Layer, a Cyber Layer, a Semantic Layer, and a Communication Layer. Each one of them is explained as follows.

3.1.1. Physical layer

The Physical Layer represents a physical thing in the physical world, which can be an actuator, a device, a system, or even a person. The

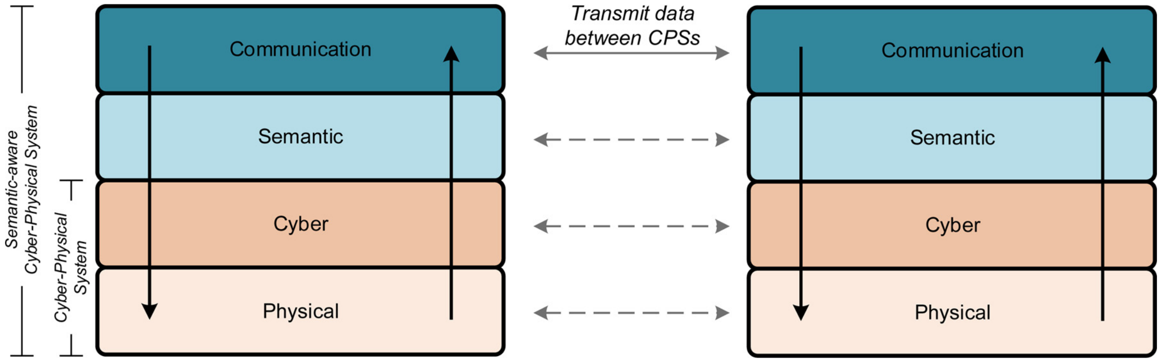


Fig. 1. An abstract system architecture of communication between two Semantic-aware CPSs.

actual thing in the Physical Layer depends on the scope of the CPS. The Physical Layer is responsible for executing physical activities in the real world, such as CNC machining and material handling. Manufacturing things can reside at different sites or organizations. Physical interactions between manufacturing things are not always possible in the future-oriented smart manufacturing network, where things are not likely in a symbiotic relationship. The network communications between manufacturing things via the Internet can establish 'logical' communications between them as if they interact with each other physically. This logic communication is required to be reliable, error-free, and time-deterministic to ensure products are produced with high precision and accuracy via deterministic manufacturing processes.

3.1.2. Cyber layer

The Cyber Layer of a manufacturing thing is tasked with sensing the behavior of the physical process, configuring its future behavior, and controlling the physical counterpart via intensive communication and feedback loops [34]. The physical model, behavior model, emotion model and trust model of a physical thing all need to be considered when an SCPS coordinates and collaborates with other objects. The Physical Layer and Cyber Layer are deeply intertwined via real-time or near real-time data synchronization through reliable wired/wireless network communication protocols. The Physical Layer and Cyber Layer form a conventional CPS.

The logical communication from one thing to another can also influence each other's behavior. A CPS can interact with other CPSs in a cooperative manner. CPSs can augment with semantics rather than merely passing data. In this way, a thing can understand the status, cognitive activities and potential actions of another manufacturing thing and collaboratively influence or control each other based on agreed goals.

3.1.3. Semantic layer

The Semantic Layer is responsible for transferring semantic messages between CPSs. It converts raw data that represents activities performed by the underlying physical thing into semantically meaningful representations and communicates with other interacting things in semantic-rich language protocols. Domain knowledge will be used to add meaning to the gathered data from the underlying CPS. Additional logical rules, such as CPS configurations policies and collaboration patterns, can also be integrated into the Semantic Layer. An inference engine can apply these rules to the knowledge base to deduce new information.

A critical component of the semantic M2M communication foundation is to establish a commonly agreed view of the domain, under which the common concepts between distributed CPSs are recognized by all the CPSs in the network. Existing domain standards and 'know-how' need to be integrated into the developed domain knowledge [35]. In this way, CPSs can communicate based on common understandings of the external working environment.

The ontological approach, which is part of the Semantic Web initiative [36], can be a useful tool for developing the Semantic Layer. Domain ontologies, semantic rules, and inference engines can work together to enable knowledge-based decision making [35,37]. The logic-based approach can also be integrated with stochastic modeling, such as machine learning algorithms, to improve the flexibility and efficiency of knowledge collection.

A Semantic Layer provides logical communication between CPSs belonging to different organizations, as shown in Fig. 2. By 'logical' communication, we mean that although communicating CPSs are not necessarily attached, from the communication viewpoint, it is as if they were connected. CPSs use the logical communication provided by the Semantic Layer to communicate, free for the worry of the details of the implementation details at the Cyber Layer.

At the sending side, the Semantic Layer converts the messages it receives from a sending CPS into semantic expressions. These semantic expressions will be further transmitted to the Communication Layer as web requests using typical network protocols, such as HTTP. At the receiving side, the Semantic Layer receives the communication requests from its Communication Layer, converts to semantic expressions and passes to the Cyber Layer to execute.

3.1.4. Communication layer

The Communication Layer serves as a direct interface between SCPSs. Semantic M2M communication is established at the Communication Layer of the interacting CPSs. Focusing on an IoT environment, complex applications, such as self-organizing networks, can be developed by employing a network of SCPSs following the practice of auto P2P communications and distributed intelligent decision-making between the networking SCPSs. The Communication Layer of an SCPS should be scalable and flexible for establishing P2P communications in an IoT environment. To this end, RESTful APIs can be one of the effective ways to structure the different types of communication patterns between interacting things over the Internet. Practically, GET requests can be used to retrieve the status of a CPS; whereas, PUT requests can be used to update the status of a CPS. In other words, a CPS can use PUT requests to control another CPS.

While communications can be successfully established between two interacting SCPSs, engineering applications that require complex communication logics at high-frequency data flow using direct P2P communications may cause communication chaos and traffic congestion because of uncoordinated arbitrary communications, unbalanced communication and data processing loads. Shared message queues could be used to facilitate asynchronous communication, group message broadcasting and traffic load control. When P2P communication becomes complex, communication congestion control and scheduling algorithms, such as [38–40] can be used to model and optimize the P2P communication network for stability, utility maximization, delay and/or throughput.

3.2. Implementation principles

Potential engineering applications of semantic-aware M2M communications pose stringent requirements on the performance of the communication technology, such as reliability, time determinism, security and privacy. The following subsections discuss these critical matters in detail.

3.2.1. Reliability

Accurate collaborative decision-making relies on reliable data transmission and network communication over the Internet. The communication reliability must increase from the Communication Layer down to the Physical Layer as the execution of collaborative automation relies on the up-to-date status of interacting SCPSs. Network communication issues such as data loss and repetition can lead to irrational manufacturing decisions. The key research issue is to design network issue detection and compensation mechanisms to ensure communication at the Cyber Layer is reliable so that zero error is guaranteed for system automation and manufacturing execution. The established methods and algorithms in the computer networks domain, such as acknowledgments, timers, retransmissions, and sequence numbers can be used to ensure communication reliability over the Internet.

3.2.2. Time determinism

Time-deterministic communication needs to be guaranteed between interacting SCPSs. The necessary implementation structure needs to guarantee that an event occurs in a specified, predictable time period. Time-deterministic or real-time communications are required for timing-critical applications, such as closed-loop motion control. Non-real-time or soft-real-time applications can be supported in some cases.

The essential requirement for guaranteeing time-deterministic communication is time-deterministic data transmission over the

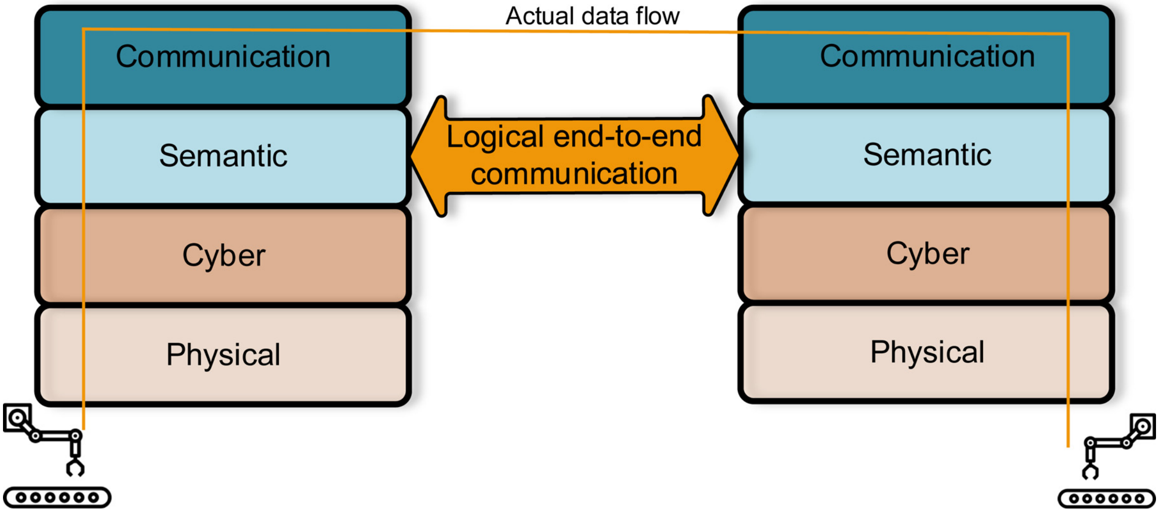


Fig. 2. Semantic Layer provides logical communication between Cyber-Physical Systems.

Internet. Existing network communication protocols, such as TCP and UDP, alone cannot guarantee time-deterministic data transmission because of random communication errors happening on a communication route [41]. Compensation mechanisms, such as dynamic data sizing [42] and buffering [43], are proposed to mitigate these errors.

Time-deterministic M2M communication is not just time-sensitive data transmission over wide area networks (WANs). Equally important, computational tasks, such as deriving adaptive control strategies between data transmission events, should also be completed with time-determinism. However, existing data processing engines might not be able to achieve deterministic latency under changing computing tasks, and data emit rate [44]. M2M communications and control over WANs will use a combination of hard, soft and non-critical tasks. Such systems will require breakthroughs on time-predictable algorithms and architecture on embedded systems, as well as time-sensitive scheduling of mixed-criticality tasks.

3.2.3. Security and privacy

In a manufacturing network, CPSs communicate and exchange data with each other as well as with their environment. However, before any communication takes place, interacting CPSs must be able to perform mutual authentication so that they can identify each other. For this purpose, a CPS is expected to get a Unique Identifier (UID) possibly certified by a trusted entity. As standardized by the OPC Foundation [14], X.509 certificates, which are certificated by a Certification Authority (CA), can be used for establishing trust in smart manufacturing environments. The CPSs can mutually authenticate each other by validating their X.509 certificates.

Authorization is also integral to secure M2M communications. With the increased concerns on cybersecurity in a networked manufacturing environment, flexible access control models are needed to regulate access to CPS status and resources. An SCPS can manage its access control policies in a networked manufacturing collaboration environment using existing mechanisms ranging from classical access control techniques [45], such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), or Role-Based Access Control (RBAC), to recent authorization technologies, such as OAuth2 [46] and OpenID Connect [47]. XACML [48] or policy-based frameworks can also be used to support more dynamic access control policies.

Access control implementation is expected to be distributed in a collaborative P2P communication network so that each SCPS can specify its accessible data by a client. A rule engine can be used to model complex data access rules. Besides, when communication mechanisms between networking manufacturing devices become truly distributed, opportunistic networks, such as Haggle [49], can be leveraged for flexible communication between SCPSs. However, this might raise privacy and security concerns for which we need to borrow ideas from existing research works, such as [50], for providing privacy and security.

4. Pilot implementation

This section reports a pilot implementation of the proposed SCPSs utilizing state-of-the-art semantic communication technologies. As discussed in Section 3, the technological focus of our proposed SCPSs architecture is upgrading a conventional CPS to an SCPS by adding a Semantic Layer and Communication Layer, with maximum compatibility with the diverse implementation architecture of a CPS. Fig. 3 depicts an implementation architecture of an SCPS using Semantic Web technologies as the key technology stack to implement the Semantic Layer and RESTful HTTP requests as the primary mechanism to enable Internet-based communication.

4.1. Semantic layer implementation

Semantic Web technologies [36], as the state-of-the-art semantic enrichment approach, have been recognized as a default method for enhancing cross-domain data semantics. Semantic Web refers to the vision of the Web of linked data. Semantic Web technologies (in Fig. 4) enable people and systems to create semantic data on the Web, build vocabulary, and write rules for handling data. As a result, machine agents can effectively search for information, inference knowledge, and make decisions on behalf of humans. Semantic Web, therefore, has been demonstrated as a powerful means of developing cognitive systems for product design [51,52], manufacturing process optimization [53–55], and cross-company manufacturing collaborations [37,56].

The Semantic Layer utilizes Semantic Web technologies to create the semantic communication interface on top of the conventional CPS. The Semantic Web-based implementation mainly contains a knowledge base-based decision-making module that can communicate in semantic languages with the support of engineering knowledge to make intelligent decisions. A Knowledge Graph is constructed by incorporating a domain ontology, engineering knowledge, and the semantic instance of the underlying CPS.

There exist several ontology development methods, one of which is the iterative ontology development approach focusing on reusing existing domain standards and ontologies [35]. In the manufacturing domain, existing manufacturing equipment modeling standards, such as

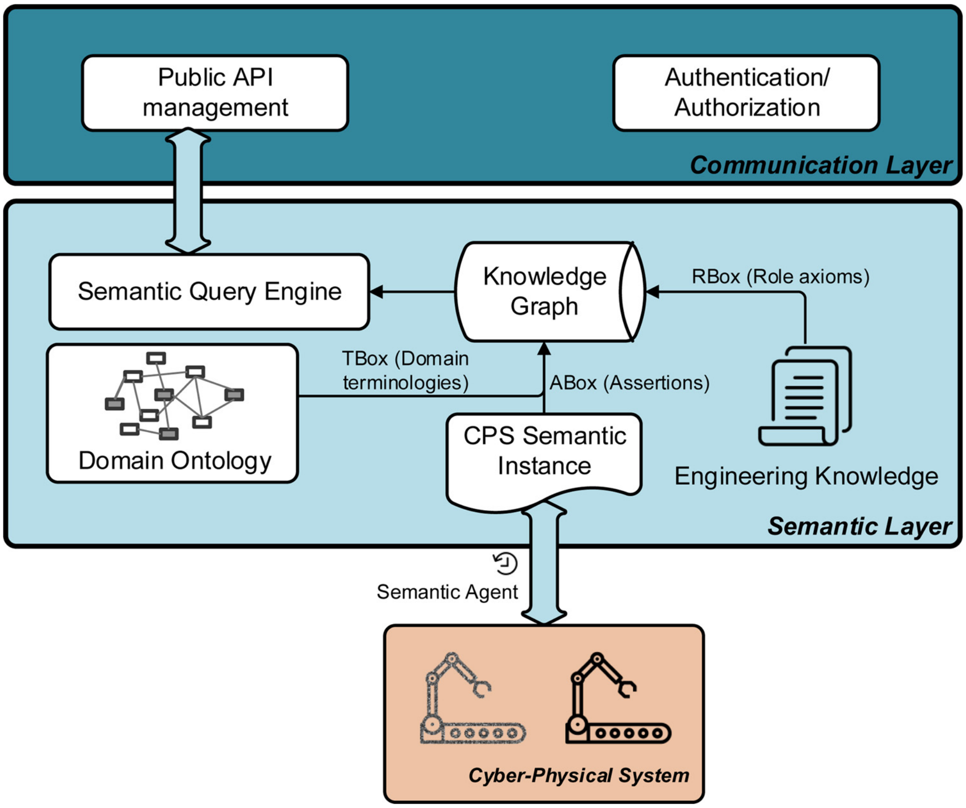


Fig. 3. Semantic Web-based implementation of SCPS.

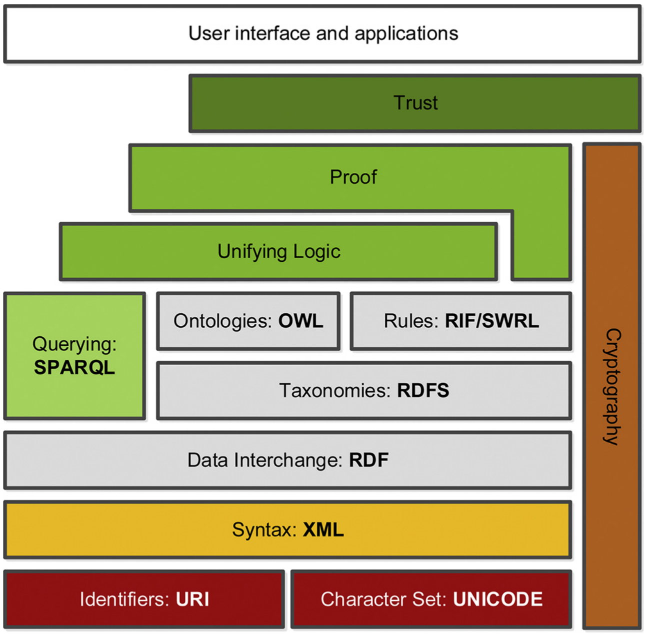


Fig. 4. The Semantic Web Stack [36].

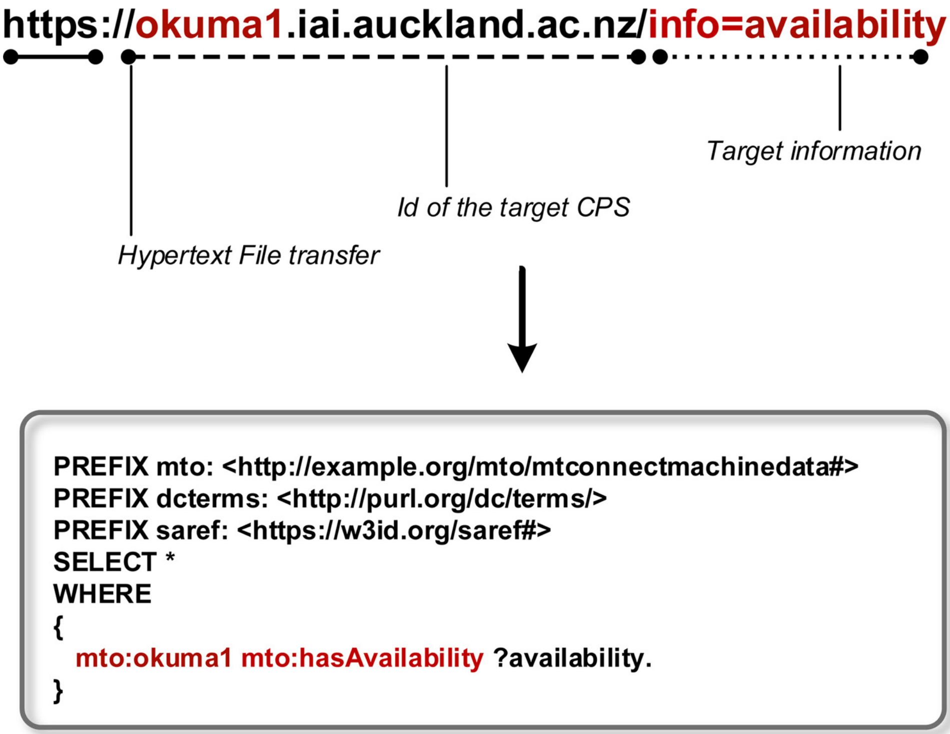
MTConnect [33] and ISO 19649 [57], need to be integrated into the developed ontology. The developed domain ontology will become a shared understanding – TBox (i.e., domain terminologies), between distributed CPSs. In this way, smart manufacturing systems can communicate with each other using a common vocabulary. The inherited interoperability of a standard-driven ontology will facilitate effortless data exchange between distributed manufacturing systems in a smart factory or manufacturing network environment [12].

The semantic instance of a CPS – ABox (i.e., Assertions) serves the purpose of updating a CPS' snapshot in the Knowledge Graph and this information will become the source of queryable CPS state, which could be exposed to the public. A Semantic Agent can be developed to synchronize a CPS' system snapshot to its ABox representation. The synchronization interval can be configured based on the possible application scenarios [58] since different applications impose different data latency requirements.

Engineering knowledge – RBox (i.e., Role axioms), can be added to the Knowledge Graph to enhance the intelligence of the Knowledge Graph so that additional knowledge about the status of the underlying CPS can be represented in a more interoperable manner to the requesters. These rules can include engineering knowledge, CPS configurations policies, collaboration patterns, and many others. A semantic inference engine can integrate these rules into the Knowledge Graph and derive new knowledge from the existing facts. The ability to infer explicit knowledge from gathered CPS data can enable cognitive communications between multiple semantic-aware CPSs.

4.2. Communication layer implementation

The implementation of the Communication Layer mainly deals with exposing the queryable status of an SCPS via HTTP requests and managing incoming communications. We adopted a RESTful architecture to encapsulate an SCPS's queryable status as consumable APIs over the Internet. The public API management module can translate an HTTP request to its equivalent representation in semantic query languages, e.g., SPARQL. To support authentication and establish a secure channel, Secure Socket Layer (SSL)/Transport Layer Security (TLS) can be used. SSL/TLS offers mutual authentication and establishes a secure channel for providing end-to-end encrypted communications over the Internet. HTTPS refers to an application-specific implementation that runs HyperText Transfer Protocol (HTTP) on top of SSL/TLS. HTTPS can be used to provide encrypted communications and secure identification of an SCPS. By default, all the elements of a Knowledge Graph



```
PREFIX mto: <http://example.org/mto/mtconnectmachinedata#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX saref: <https://w3id.org/saref#>
SELECT *
WHERE
{
  mto:okuma1 mto:hasAvailability ?availability.
}
```

Fig. 5. Conversion from HTTP requests to semantic queries.

are queryable, including entities, individuals, object properties and data properties. As summarized in [45], existing access control models work well with SPARQL for regulating access over what can be queried. In particular, DAC [59], MAC [60], RBAC [61] and XACML [62] are access mechanisms that can be used. More specifically, the authorization rules by these access control mechanisms can define what can be requested and retrieved by an SCPS.

Then the Public API Management Module needs to create a complete list of mappings between valid HTTP requests and their equivalent semantic queries in SPARQL expressions. A sample conversion from an HTTP GET request to its SPARQL expression is shown in Fig. 5. The HTTP GET request specifies the CPS ID and the target information to fetch. The ID of a CPS adopts the Domain Name System (DNS) to ensure unique CPSs are listed over the Internet. For example, in this example, 'okuma1' is a subdomain that is assigned to a unique CPS under the subdomain of 'iai.auckland.ac.nz'. Compliance with DNS can ensure a CPS is uniquely registered over the Internet across organizations. In the case of Fig. 5, the HTTP GET request will be converted to a SPARQL query that fetches the availability information of 'okuma1.iai.auckland.ac.nz'.

4.3. Case studies

To validate the proposed SCPSs, we created a test application to examine semantic M2M communications between multiple machine tools at the Smart Manufacturing System Testbed [63] developed by the US National Institutes of Standards and Technology (NIST). This testbed offers MTConnect-based machine tool monitoring services, upon which we implemented the Semantic Layer and Communication Layer for Semantic Web-based communication.

The NIST Testbed adopts MTConnect standard as the information model for representing connected machine tools. As discussed in Section 4.1, we first converted the MTConnect standard to an ontology, based on the work in [64], whose main concepts were represented in Fig. 6. A Semantic Agent was developed to synchronize machine tool snapshots between the underlying MTConnect-compatible machine tool and the Semantic Layer. The Semantic Agent monitors the status of the underlying machine tool and updates its ABox in the Knowledge Graph. The implementation of the Semantic Agent is a client application in the MTConnect reference architecture. The pseudocode of the semantic representation synchronization algorithm for MTConnect-compatible machine tools is illustrated in Algorithm 1. The algorithm starts with the initiation of a complete machine tool snapshot in the Knowledge Graph. The Semantic Agent then subscribes to machine tool data change and specifies the data update interval and heartbeat rate (both are standard terms in MTConnect specifications). Subsequent machine tool data item changes will trigger an update to the machine tool's semantic representation in the Knowledge Graph.

Algorithm 1: Synchronize machine tool snapshot as semantic representation

```
Data: MTConnect-compatible machine tool end point
Result: Synchronized machine tool semantic representation

initialization;
if ABox representation does not exist then
    Fetch machine tool CURRENT data;
    Create ABox instance, save to database;
end
Subscribe to MTConnect Agent property changes, set messaging intervals and
  heartbeat;
if Machine tool property changes then
    Update ABox instance, save to database;
end
```

In this case study, we upgraded all the available machine tools at NIST Testbed to SCPSs to test the proposed semantic communication mechanism. Fig. 7 represents a snippet of the ABox representation of the Mazak machine in the turtle format.

Fig. 8 illustrates the communication processes between an MTConnect-compatible semantic-aware machine tool and its clients. First, a client establishes a secure communication channel with the SCPS. This client can be any external entity, such as another SCPS. The client interacts with the Communication Layer of the SCPS. The process starts with a handshake in which the client sends a CLIENT HELLO message. The SCPS responds with a SERVER HELLO message. This completes the negotiation of the protocol version, session ID, cipher suites, and compression methods. Next, they perform mutual

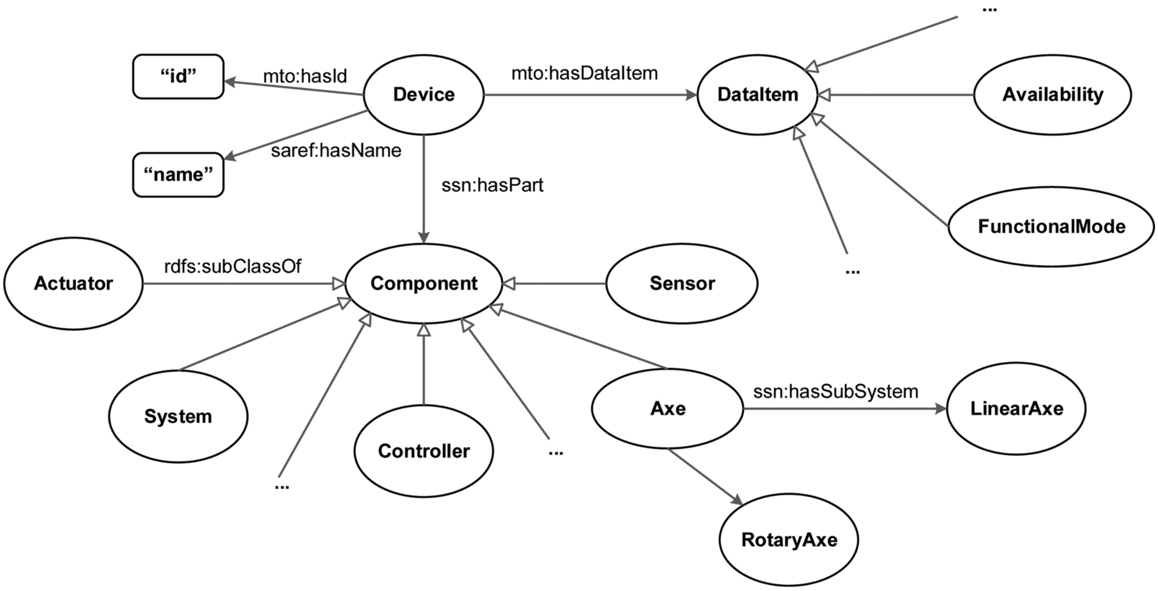


Fig. 6. Graphical representation of the MTConnect main concepts [64].

authentication and exchange certificate based on the key exchange method supported by both. To this end, one possibility is to perform mutual authentication using X.509 certificates. Then, a session key is established. Using this session key, both parties can securely exchange messages, which can be encrypted and tamper-resistant, thus ensuring confidentiality and integrity, respectively.

The client can then send queries to the Communication Layer, which performs authorization checks to decide whether the client can make those queries to get the requested data. Once authorized, the Communication Layer will translate the RESTful query into a SPARQL query against the Knowledge Graph in the Semantic Layer. The Semantic Layer will also check whether the machine tool snapshot in the Semantic Layer has expired or not. If the snapshot has expired, the Semantic Layer will fetch a new snapshot from the underlying Cyber Layer using MTConnect GET requests. The new snapshot will be added to the Knowledge Graph and returned to the client. If the stored machine tool snapshot has not expired, the Semantic Layer will return the stored snapshot to the client. In the meantime, the Semantic Agent continuously monitors changes from the physical machine tool and updates the machine semantic snapshot whenever a machine tool status change is detected. M2M communications were tested on the following two scenarios.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn> .
@prefix mto: <http://example.org/mto/mtconnectmachinedata#> .
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix part: <http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/
part.owl#>.
@prefix saref: <https://w3id.org/c>.
@prefix sem: <http://semanticweb.cs.vu.nl/2009/11/sem/>.

mto:Nexus1
        rdfs:label "Mazak03";
        rdf:type ssn:Device, owl:NamedIndividual;
        rdfs:isDefinedBy mto: ;
        mto:hasCreationtime "2018-02-11T19:29:34Z"^^xsd:dateTime;
        mto:hasId "Mazak03";
        saref:hasName "Mazak03";
        saref:hasManufacturer "Mazak";
        saref:hasModel "QTN";
        mto:hasUuid "mtc_adapter004";
        saref:hasDescription  "Mazak QuickTurn - Mazak QuickTurn Nexus 300";
        ssn:hasPart mto:Axe1, mto:Controller1, mto:System1;
        mto:hasDataItem mto:Availability1, mto:AssetChanged, mto:AssetRemoved.
mto:Availability1
                 rdf:type mto:Availability ;
                 mto:hasId "Mazak03-dtop_1";
                 mto:hasName "avail";
                 mto:hasSequence "8093061";
                 sem:hasTimeStamp "2018-02-11T13:50:03.871849";
                 mto:hasAvailability "AVAILABLE";
                 mto:hasCategory "EVENT".

mto:AxeActuatorCon
                rdf:type mto:ActuatorCondition ;
                saref:hasName "servo_cond";
                mto:hasId "Mazak03-base_2";
                mto:hasSequence "8093062";
                sem:hasTimeStamp "2018-02-11T13:50:04.771668";
                mto:hasCategory "CONDITION";
                mto:hasCondition "NORMAL".

mto:Controller1
        rdf:type mto:Controller;
        rdfs:isDefinedBy mto: ;
        part:partOf mto:I400;
        owl:differentFrom mto:Axe1, mto:System1;
        mto:hasId "Mazak03-controller_1";
        saref:hasName "controller";
        ssn:hasSubSystem mto:Path1;
        mto:hasDataItem mto:ConCommunication1, mto:ConLogicProgram1,
mto:SystemCondition1, mto:ConPalletNumber1.
mto:ConCommunication1 rdf:type mto:Communication;
                saref:hasName "comms_cond";
                mto:hasId "Mazak03-controller_2";
                mto:hasSequence "8093063";
                sem:hasTimeStamp "2018-02-11T13:50:04.771791";
                mto:hasCondition "NORMAL";
                mto:hasCategory "CONDITION".
mto:ConPalletNumber1 rdf:type mto:PalletNumber;
                saref:hasName "pallet_num";
                mto:hasId "Mazak03-controller_5";
                mto:hasSequence "8093072";
                sem:hasTimeStamp "2018-02-11T13:50:04.772536";
                mto:hasPalletNumber 0;
                mto:hasCategory "EVENT".
mto:SystemCondition1 rdf:type mto:SystemCon;
                saref:hasName "system_cond";
                mto:hasId "Mazak03-controller_4";
                mto:hasSequence "8133808";
                sem:hasTimeStamp "2018-02-11T17:24:34.271743";
                mto:hasCondition "NORMAL";
                mto:hasCategory "CONDITION".
mto:ConLogicProgram1 rdf:type mto:LogicProgram;
                saref:hasName "logic_cond";
                mto:hasId "Mazak03-controller_3";
                mto:hasSequence "8093064";
                sem:hasTimeStamp "2018-02-11T13:50:04.771897";
                mto:hasCondition "NORMAL";
                mto:hasCategory "CONDITION".
```

Fig. 7. A snippet of the ABox representation of the Mazak machine in the NIST Testbed in the turtle format.

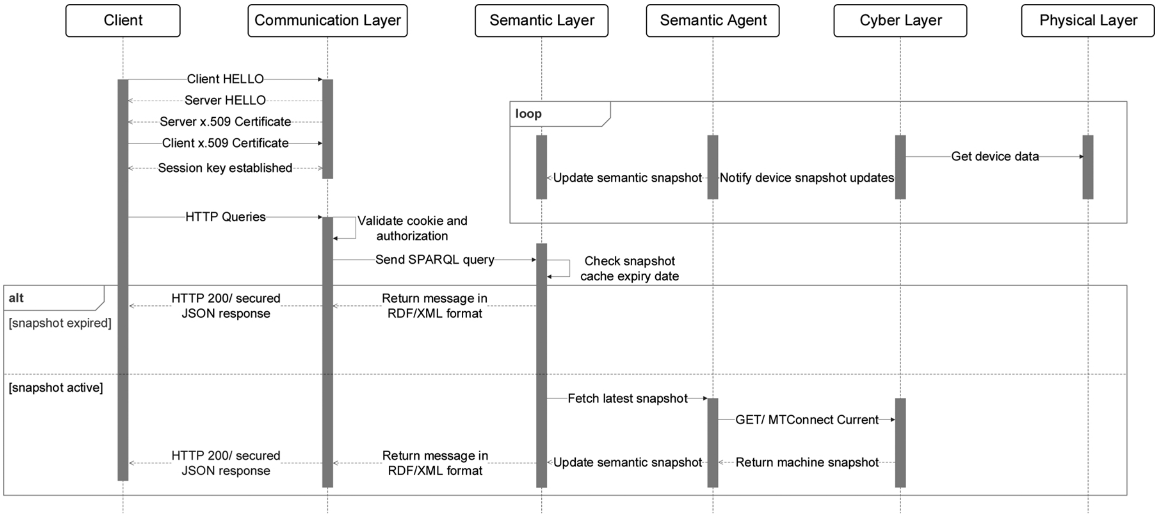


Fig. 8. Process flow of communication with MTConnect-compatible semantic-aware machine tools.

Context: Multiple machines are working collaboratively on a manufacturing job with each machine working on part of the job. A machine needs to monitor the working state of other machines and the manufacturing job.

Objective: The Hurco machine fetches the availability information of all the machines and their current working-in-progress manufacturing jobs.

Based on the above scenario, Hurco can send a SPARQL query to all the connected machines to get their availability. The query result returns a list of connected machines owned by ‘iai.auckland.ac.nz’ and their availability information. With some further data manipulation, the Web API returns the data in JSON format, as shown in Fig. 9.

The second case demonstrates the capability of knowledge-based semantic communication, where additional knowledge can be incorporated to return better context-aware information.

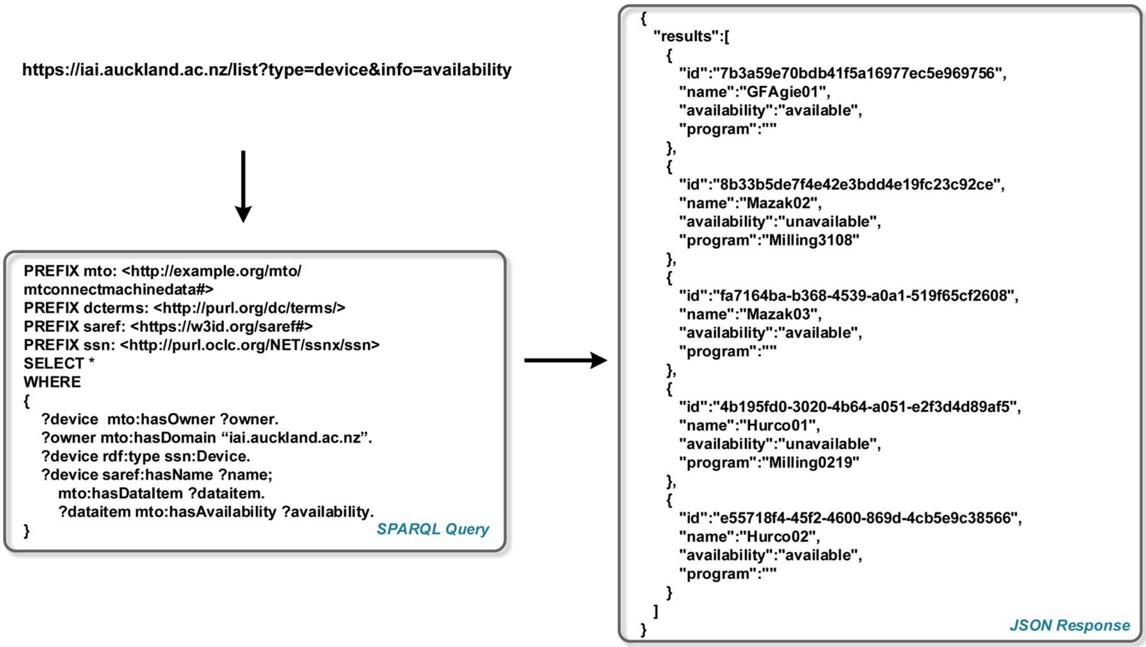
Context: A smart factory has a centralized library of cutters that are shared between multiple CNC machines in the factory. A CNC machine needs to predictively schedule its required cutters for fulfilling its upcoming manufacturing tasks to minimize production downtime. As such, a CNC machine is required to be capable of obtaining the status of each cutter in the cutter library.

Objective: A machine obtains the real-time availability information of a cutter from the CNC machines in a factory.

The above objective can be achieved by tracking all the in-use cutters on all the machine tools in the factory. This data can be used to predict the availability of each cutter based on the production progress of its associated production job. To return more context-aware information, the following Jena rule, as specified in Fig. 10 were added to the Knowledge Graph to assist with deriving context-aware information. This rule states that a cutter is not available if it is being used by an unavailable machine tool.

Based on the above engineering rule, an HTTP request that fetches the availability of all the cutters can return all the cutters with their availability information, as shown in Fig. 11.

We further enriched the above implementation in the Laboratory of Industry 4.0 Smart Manufacturing Systems at the University of Auckland to demonstrate the benefits and working mechanisms of semantic-aware M2M communications for collaborative smart



https://iai.auckland.ac.nz/list?type=device&info=availability

```
PREFIX mto: <http://example.org/mto/
mtconnectmachinedata#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX saref: <https://w3id.org/saref#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn>
SELECT *
WHERE
{
   ?device  mto:hasOwner ?owner.
   ?owner mto:hasDomain “iai.auckland.ac.nz”.
   ?device rdf:type ssn:Device.
   ?device saref:hasName ?name;
      mto:hasDataItem ?dataitem.
      ?dataitem mto:hasAvailability ?availability.
}
```

```
{
  "results":[
    {
      "id":"7b3a59e70bdb41f5a16977ec5e969756",
      "name":"GFAgie01",
      "availability":"available",
      "program":""
    },
    {
      "id":"8b33b5de7f4e42e3bdd4e19fc23c92ce",
      "name":"Mazak02",
      "availability":"unavailable",
      "program":"Milling3108"
    },
    {
      "id":"fa7164ba-b368-4539-a0a1-519f65cf2608",
      "name":"Mazak03",
      "availability":"available",
      "program":""
    },
    {
      "id":"4b195fd0-3020-4b64-a051-e2f3d4d89af5",
      "name":"Hurco01",
      "availability":"unavailable",
      "program":"Milling0219"
    },
    {
      "id":"e55718f4-45f2-4600-869d-4cb5e9c38566",
      "name":"Hurco02",
      "availability":"available",
      "program":""
    }
  ]
}
```

Fig. 9. An example of fetching the availability information of machine tools.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix mto: <http://example.org/mto/mtconnectmachinedata#>
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn>
@prefix saref: <https://w3id.org/saref#>

[(?cutter mto:hasAvailability “unavailable”) <-
(?machineTool ssn:hasPart ?controller),
(?machineTool rdf:type ssn:Device),
(?machineTool mto:hasDataItem ?dataitem),
(?dataitem mto:hasAvailability ?availability),
equal(?availability, “AVAILABLE”),
(?controller rdf:type mto:Controller),
(?controller ssn:hasSubSystem, ?path),
(?path rdf:type mto:path),
(?path mto:hasDataItem ?toolAssetId),
(?toolAssetId rdf:type mto:ToolAssetId),
(?toolAssetId mto:hasToolAssetId ?cutter)]
```

Fig. 10. Cutter availability rules in Jena syntax.

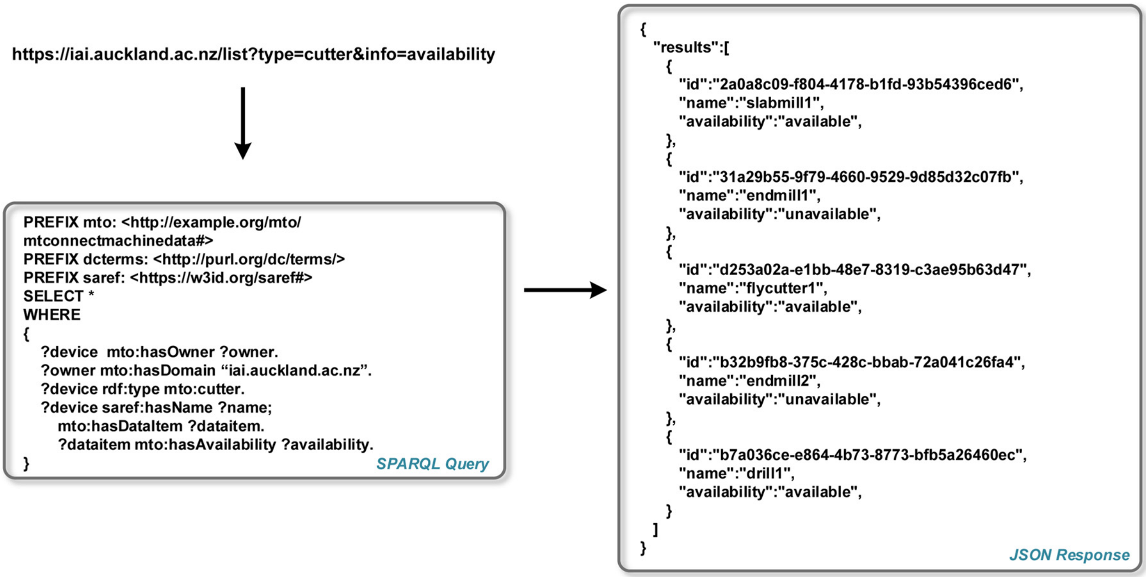
manufacturing automation. Fig. 12 shows a workshop that is equipped with semantic-aware M2M technologies. The workshop includes a variety of manufacturing devices, such as a milling machine, a lathe, a KUKA drilling robot, a conveyor and a KUKA material transfer unit. We implemented decentralized data-driven decision-making between the manufacturing devices – each manufacturing device makes its own decision on its upcoming action based on its own control strategy and other manufacturing devices' status. P2P semantic communication is available as all the manufacturing devices are retrofitted as semantic-aware CPSs, following the pilot implementation introduced in this research. The workshop can produce specialized mechanical seal shaft sleeves. The manufacturing processes are mainly turning and milling operations and the KUKA Drilling Robot is used on-demand to drill the locating hole on the shaft sleeve to avoid tool changes on the CNC machines for drilling operations. In this case study, the KUKA Drilling Robot periodically requests the job status of the current job on the lathe via an HTTP endpoint, similar to that in Figs. 9 and 11. Based on the machining progress of the current job and predicted remaining machining time, the KUKA Drilling Robot can decide the timing that the KUKA Material Transfer Unit starts transporting the semi-finished workpiece to the conveyor and the timing that the drilling robot needs to execute its drilling operations for machining the locating hole. In addition, the KUKA Material Transfer Unit between the KUKA Drilling Robot and the Lathe can also detect the job arrival rate from the Lathe and then automatically suggests the drilling operation speed of the KUKA Drilling Robot to ensure stable operation outputs from the KUKA Drilling Robot. This distributed production scheduling and control based on semantic-aware M2M communications increases production process flexibility and productivity compared to traditional centralized supervision control.

5. Discussions and research directions

The research outcome demonstrated the feasibility of integrating CPSs with Semantic Web technologies to enable semantic interactions between manufacturing systems. Case studies showed the preliminary implementation of the technologies on two networked machine tools, which enabled two machine tools at different locations to query each other's status via a semantic web query language. The research work will enable distributed manufacturing systems to establish P2P communications and develop intelligent smart manufacturing applications, such as knowledge-based self-configuration and collaborative manufacturing automation. Nevertheless, industry practitioners that are willing to deploy the solution to industry applications would need to integrate functionalities common to industrial systems, such as authentication and authorization, to mitigate security risks.

Offering authentication using digital certificates in HTTPS can be challenging for a resource-constrained SCPS because validating the revocation status of those certificates incurs a high overhead. To address this concern, investigating light-weight solutions to validate digital certifications could be a possibility [65]. One might argue that connection-oriented protocols, such as TCP, might be cumbersome for an SCPS. For this purpose, we might consider QUIC [66], which is faster as it is based on UDP as well as it achieves some TCP properties, such reliability. In other words, QUIC can offer security and efficiency at the same time for SCPS.

More flexible approaches need to be used to achieve dynamic authorization policies. For instance, a CPS container can define what can be accessed (i.e., a whitelist approach), what cannot be accessed (i.e., a blacklist approach), or a combination of both (i.e., a hybrid approach). In the case of hybrid authorization policies, conflict resolution strategies might be required, which may take into account domain-specific knowledge. A CPS can get faulty or start behaving maliciously. To this end, the critical challenge is to develop novel approaches that can



```
https://iai.auckland.ac.nz/list?type=cutter&info=availability
```

```
PREFIX mto: <http://example.org/mto/
mtconnectmachinedata#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX saref: <https://w3id.org/saref#>
SELECT *
WHERE
{
   ?device  mto:hasOwner ?owner.
   ?owner mto:hasDomain “iai.auckland.ac.nz”.
   ?device rdf:type mto:cutter.
   ?device saref:hasName ?name;
      mto:hasDataItem ?dataitem.
      ?dataitem mto:hasAvailability ?availability.
}
```

```
{
  "results":[
    {
      "id":"2a0a8c09-f804-4178-b1fd-93b54396ced6",
      "name":"slabmill1",
      "availability":"available",
    },
    {
      "id":"31a29b55-9f79-4660-9529-9d85d32c07fb",
      "name":"endmill1",
      "availability":"unavailable",
    },
    {
      "id":"d253a02a-e1bb-48e7-8319-c3ae95b63d47",
      "name":"flycutter1",
      "availability":"available",
    },
    {
      "id":"b32b9fb8-375c-428c-bbab-72a041c26fa4",
      "name":"endmill2",
      "availability":"unavailable",
    },
    {
      "id":"b7a036ce-e864-4b73-8773-bfb5a26460ec",
      "name":"drill1",
      "availability":"available",
    }
  ]
}
```

Fig. 11. An example of fetching the availability information of cutters with the assistance of engineering rules.

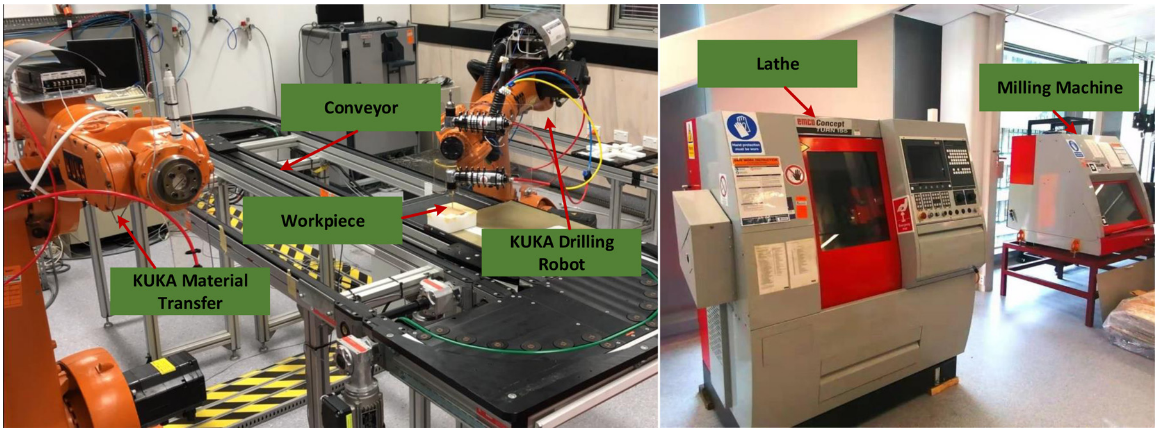


Fig. 12. Demonstrative collaborative manufacturing automation setup based on semantic-aware communication technologies.

identify faulty and misbehaving CPSs in their early stages, possibly by involving benign CPSs in decision-making processes.

The wide-scale adoption of semantic interactions between manufacturing systems needs widely-accepted standards in place, such as MTConnect used in this research. We believe the strong research activities on Industry 4.0 standardization will lay the foundation for ubiquitous communications between systems. Two possible P2P communication scenarios can co-exist in the future as a result of the development of Industry 4.0 standards. One scenario is that all interacting CPSs can communicate via one single neutral domain ontology that encompasses all the possible concepts and relations within the industry domain, with which all the possible manufacturing things can be modeled with standard interfaces. The dream of a one-fits-all ontology can significantly reduce the cost of M2M communication and collaboration. However, a bottom-up approach can speed up the pace of standardization. This represents another more complicated scenario where partial ‘bridging ontologies’ exist between a group of CPSs for facilitating information exchange. MTConnect standard is such an example.

Another enabler of semantic-aware CPS is control between CPSs over the Internet. The unreliable network communications over the Internet conflicts with the reliability, accuracy and time determinism that are required by industrial control. Research needs to focus on the separation and integration of unreliable networks with reliable industrial control in one system with the assistance of technologies that intelligently schedule, coordinate and monitor mixed-criticality tasks.

6. Conclusions and future work

Aiming at developing the technological foundation for collaborative smart manufacturing automation, we proposed the concept of Semantic-aware CPS and developed the enabling technologies for establishing semantic communications between SCPSs. To the best of our knowledge, this is the first piece of work that considers semantic-aware M2M communications from the CPS development point of view to create the foundation for ubiquitous communication and collaboration between distributed manufacturing things. Based on this work, future manufacturing systems can evolve into a smart manufacturing network that can self-organize and self-optimize its configurations to adapt to dynamic operating conditions.

Our primary research contribution lies in the proposal of a generic architecture of Semantic-aware CPSs based on Semantic Web technologies, using which CPSs can establish semantic communications as needed. The proposed SCPS adopts a layered architecture that separates the implementation of conventional CPS and semantic communications, which makes it compatible with diverse CPS implementation architectures. This significantly enables the smooth upgrade of a CPS to an SCPS. Our communication study between multiple machine tools over the Internet demonstrated the capability of the proposed SCPSs for empowering P2P M2M communications.

We believe the concept of SCPS can inspire in-depth research on the infrastructure of the future smart manufacturing. Future research work will be on enhancing the implementation of the proposed system and developing more plug-and-play semantic agents for different types of CPSs. Besides, validating the concept of SCPS in more complex application settings could be another focus.

Declaration of Competing Interest

No conflict of interest.

References

[1] Davis J, Edgar T, Porter J, Bernaden J, Sarli M. Smart manufacturing, manufacturing intelligence and demand-dynamic performance. Comput Chem Eng 2012;47:145–56. https://doi.org/10.1016/J.COMPCHEMENG.2012.06.037.
[2] Wang L, Shih AJ. Challenges in smart manufacturing. J Manuf Syst 2016;40:1. https://doi.org/10.1016/j.jmsy.2016.05.005.
[3] Kagermann H, Helbig J, Hellinger A, Wahlster W. Recommendations for implementing the strategic initiative INDUSTRIE 4.0: securing the future of german manufacturing industry; final report of the industrie 4.0 working group. Forschungsunion 2013.
[4] Lu Y, Xu X. Resource virtualization: a core technology for developing cyber-physical production systems. J Manuf Syst 2018;47:128–40. https://doi.org/10.1016/j.jmsy.2018.05.003.
[5] Meng Z, Wu Z, Gray J, Meng Z, Wu Z, Gray J. A collaboration-oriented M2M messaging mechanism for the collaborative automation between machines in future industrial networks. Sensors 2017;17:2694. https://doi.org/10.3390/s17112694.
[6] Wollschlaeger M, Sauter T, Jasperneite J. The future of industrial communication: automation networks in the era of the internet of things and industry 4.0. IEEE Ind Electron Mag 2017;11:17–27. https://doi.org/10.1109/MIE.2017.2649104.
[7] Willner A, Diedrich C, Ben Younes R, Hohmann S, Kraft A. Semantic communication between components for smart factories based on oneM2M. 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) 2017. p. 1–8. https://doi.org/10.1109/ETFA.2017.8247690.
[8] Monostori L. Cyber-physical production systems: roots, expectations and R&D challenges. Procedia CIRP 2014;17:9–13. https://doi.org/10.1016/j.procir.2014.03.115. Elsevier B.V.
[9] Tantik E, Anderl R. Integrated data model and structure for the asset administration shell in industrie 4.0. Procedia CIRP 2017;60:86–91. https://doi.org/10.1016/j.procir.2017.01.048.
[10] Mittal S, Khan MA, Romero D, Wuest T. Smart manufacturing: characteristics, technologies and enabling factors. Proc Inst Mech Eng Part B J Eng Manuf 2017. https://doi.org/10.1177/0954405417736547. 095440541773654.
[11] Stankovic JA. Research directions for the internet of things. IEEE Internet Things J 2014;1:3–9. https://doi.org/10.1109/JIOT.2014.2312291.
[12] Lu Y, Liu C, Wang KI-K, Huang H, Xu X. Digital twin-driven smart manufacturing: connotation, reference model, applications and research issues. Robot Comput Integr Manuf 2020;61:101837https://doi.org/10.1016/j.rcim.2019.101837.
[13] Rowley J. The wisdom hierarchy: representations of the DIKW hierarchy. J Inf Sci 2007;33:163–80. https://doi.org/10.1177/0165551506070706.
[14] OPC Foundation. Unified architecture. OPC Foundation; 2019 n.d. https://opcfoundation.org/about/opc-technologies/opc-ua/ (Accessed 25 April 2019).
[15] Miyazawa I, Murakami M, Matsukuma T, Fukushima K, Maruyama Y, Matsumoto

M, et al. OPC UA information model, data exchange, safety and security for IEC 61131–3. SICE Annual Conference 2011. 2011. p. 1556–9.

[16] Trnka P, Kodet P, Havlena V. OPC-UA information model for large-scale process control applications. IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society. 2012. p. 5793–8.

[17] Maka A, Cupek R, Rosner J. OPC UA object oriented model for public transportation system. 2011 UKSim 5th European Symposium on Computer Modeling and Simulation 2011. p. 311–6. https://doi.org/10.1109/EMS.2011.84.

[18] Sobel W. MTConnect standard. Part 1—overview and protocol. 2015. p. 10.

[19] Edrington B, Zhao B, Hansel A, Mori M, Fujishima M. Machine monitoring system based on MTConnect technology. Procedia CIRP 2014;22:92–7. https://doi.org/10.1016/J.PROCIR.2014.07.148.

[20] Shin S-J, Woo J, Kim DB, Kumaraguru S, Rachuri S. Developing a virtual machining model to generate MTConnect machine-monitoring data from STEP-NC. Int J Prod Res 2016;54:4487–505. https://doi.org/10.1080/00207543.2015.1064182.

[21] Liu C, Vengayil H, Lu Y, Xu X. A cyber-physical machine tools platform using OPC UA and MTConnect. J Manuf Syst 2019;51:61–74. https://doi.org/10.1016/j.jmsy.2019.04.006.

[22] Lee B, Kim D-K, Yang H, Oh S. Model transformation between OPC UA and UML. Comput Stand Interfaces 2017;50:236–50.

[23] MTConnect Institute. OPC UA companion specification. 2019https://www.mtconnect.org/opc-ua-companion-specification.

[24] Henssen R, Schleipen M. Interoperability between OPC UA and AutomationML. Procedia CIRP 2014;25:297–304.

[25] Maracic H, Miskovic T, Kusek M, Lovrek I. Context-aware multi-agent system in machine-to-machine communication. Procedia Comput Sci 2014;35:241–50. https://doi.org/10.1016/J.PROCS.2014.08.104.

[26] Grangel-Gonzalez I, Halilaj L, Coskun G, Auer S, Collarana D, Hoffmeister M. Towards a semantic administrative shell for industry 4.0 components. 2016 IEEE Tenth International Conference on Semantic Computing (ICSC) 2016. p. 230–7. https://doi.org/10.1109/ICSC.2016.58.

[27] Grangel-Gonzalez I, Halilaj L, Auer S, Lohmann S, Lange C, Collarana D. An RDF-based approach for implementing industry 4.0 components with administration shells. 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA) 2016. p. 1–8. https://doi.org/10.1109/ETFA.2016.7733503.

[28] Kirkizh V, Komar M, Petrov V. Towards semantic web: seamless integration of services and devices for the FRUCT community. Conference of Open Innovation Association, FRUCT, 2013. 2017. p. 53–62. https://doi.org/10.23919/FRUCT.2013.8124229.

[29] Mazayev A, Martins JA, Correia N. Interoperability in IoT through the semantic profiling of objects. IEEE Access 2017;6:19379–85. https://doi.org/10.1109/ACCESS.2017.2763425.

[30] Grangel-González I. Semantic data integration for industry 4.0 standards. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10180. LNAI, Springer Verlag; 2017. p. 230–7. https://doi.org/10.1007/978-3-319-58694-6_36.

[31] Lee J, Bagheri B, Kao H-A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manuf Lett 2015;3:18–23. https://doi.org/10.1016/J.MFGLET.2014.12.001.

[32] The Reference Architectural Model Industrie 4.0 (RAMI 4.0) - zvei.org n.d. https://www.zvei.org/en/press-media/publications/the-reference-architectural-model-industrie-40-rami-40/. (Accessed 8 June 2019).

[33] MTConnect Institute. Standard — MTConnect n.d. https://www.mtconnect.org/standard20181. (Accessed 24 April 2019).. 2019.

[34] Wang L, Törngren M, Onori M. Current status and advancement of cyber-physical systems in manufacturing. J Manuf Syst 2015;37:517–27. https://doi.org/10.1016/j.jmsy.2015.04.008.

[35] Lu Y, Wang H, Xu X. ManuService ontology: a product data model for service-oriented business interactions in a cloud manufacturing environment. J Intell Manuf 2019;30:317–34. https://doi.org/10.1007/s10845-016-1250-x.

[36] Berners-Lee T, Hendler J, Lassila O. The semantic web. Sci Am 2001;284:34–43. doi:1.

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

[38] Ephremides A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Trans Automat Contr 1992;37:1936–48. https://doi.org/10.1109/9.182479.

[39] Neely MJ, Modiano E, Li CP. Fairness and optimal stochastic control for heterogeneous networks. IEEE/Acm Trans Netw 2008;16:396–409. https://doi.org/10.1109/TNET.2007.900405.

[40] Neely MJ. Stochastic network optimization with application to communication and queueing systems. Morgan & Claypool; 2010.

[41] Schlechtendahl J, Kretschmer F, Sang Z, Lechler A, Xu X. Extended study of network capability for cloud based control systems. Robot Comput Integr Manuf 2017;43:89–95. https://doi.org/10.1016/J.RCIM.2015.10.012.

[42] Schlechtendahl J, Kretschmer F, Lechler A, Verl A. Communication mechanisms for cloud based machine controls. Procedia CIRP 2014;17:830–4. https://doi.org/10.1016/j.procir.2014.01.074. Windsor, ON: Elsevier B.V.

[43] Kretschmer F, Friedl S, Lechler A, Verl A. Communication extension for cloud-based machine control of simulated robot processes. Proceedings of the IEEE International Conference on Industrial Technology, vol. 2016. 2016. p. 54–8. https://doi.org/10.1109/ICIT.2016.7474725.

[44] Chintapalli S, Dagit D, Evans B, Farivar R, Graves T, Holderbaugh M, et al. Benchmarking streaming computation engines: storm, flink and spark streaming. 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) 2016. p. 1789–92. https://doi.org/10.1109/IPDPSW.2016.138.

[45] Kirrane S, Mileo A, Decker S. Access control and the resource description framework: a survey. Semant Web 2016;8:311–52. https://doi.org/10.3233/SW-160236.

[46] Hardt D. The OAuth 2.0 authorization framework. 2012.

[47] Sakimura N, Bradley J, Jones M, de Medeiros B, Mortimore C. OpenID connect core 1.0. Specification. OpenID Foundation; 2014. n.d.

[48] Anderson A, Nadalin A, Parducci B, Engovatov D, Lockhart H, Kudo M, et al. Extensible access control markup language (xacml) version 1.0. OASIS. 2003.

[49] Scott J, Crowcroft J, Hui P, Diot C. Haggle: A networking architecture designed around mobile users. 2006.

[50] Asghar MR, Gehani A, Crispo B, Russello G. PIDGIN: privacy-preserving interest and content sharing in opportunistic networks. Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security - ASIA CCS' 14 2014. p. 135–46. https://doi.org/10.1145/2590296.2590303.

[51] Zhao W, Liu JK. OWL/SWRL representation methodology for EXPRESS-driven product information model. Part I. Implementation methodology. Comput Ind 2008;59(580). doi:1.

[52] Aziz H, Gao J, Maropoulos P, Cheung WM. Open standard, open source and peer-to-peer tools and methods for collaborative product development. Comput Ind 2005;56:260–71.

[53] Šormaz D, Sarkar A. SIMPM–upper-level ontology for manufacturing process plan network generation. Robot Comput Integr Manuf 2019;55:183–98.

[54] Cao Q, Giustozzi F, Zanni-Merk C, de Bertrand de Beuvron F, Reich C. Smart condition monitoring for industry 4.0 manufacturing processes: an ontology-based approach. Cybern Syst 2019;50:82–96. https://doi.org/10.1080/01969722.2019.1565118.

[55] Liang JS. An ontology-oriented knowledge methodology for process planning in additive layer manufacturing. Robot Comput Integr Manuf 2018;53:28–44. https://doi.org/10.1016/J.RCIM.2018.03.003.

[56] Lin HK, Harding JA. A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. Comput Ind 2007;58:428–37. https://doi.org/10.1016/j.compind.2006.09.015.

[57] ISO/TC 299. ISO 19649:2017, Mobile robots - Vocabulary 2017.

[58] Rauchhaupt L, Kadel G, Mildner F, Kärcher B, Heidel R, Graf U, et al. Network-based communication for industries 4.0-Proposal for an administration shell. 2016.

[59] Kirrane S, Abdelrahman A, Mileo A, Decker S. Secure manipulation of linked data. International Semantic Web Conference. Springer; 2013. p. 248–63.

[60] Kodali N, Farkas C, Wijesekera D. An authorization model for multimedia digital libraries. Int J Digit Libr 2004;4:139–55.

[61] Calero JMA, Perez GM, Skarmeta AFG. Towards an authorisation model for distributed systems based on the Semantic Web. Iet Inf Secur 2010;4:411–21.

[62] Franzoni S, Mazzoleni P, Valtolina S, Bertino E. Towards a fine-grained access control model and mechanisms for semantic databases. IEEE International Conference on Web Services (ICWS 2007). 2007. p. 993–1000.

[63] Hedberg Jr. T, Helu M. Design and configuration of the smart manufacturing systems test bed. Gaithersburg, MD 2017. https://doi.org/10.6028/NIST.AMS.200-1.

[64] Alvanou G, Lytra I, Petersen N. An MTConnect ontology for semantic industrial machine sensor analytics. CEUR Workshop Proceedings 2018;vol. 2112:56–62.

[65] Hu Q, Asghar MR, Brownlee N. Checking certificate revocation efficiently using certificate revocation guard. J Inf Secur Appl 2019;48:102356.

[66] Carlucci G, De Cicco L, Mascolo S. HTTP over UDP: an experimental investigation of QUIC. Proceedings of the 30th Annual ACM Symposium on Applied Computing, ACM 2015:609–14.