Ontology for manufacturing resources in a cloud environment

Yuqian Lu and Qun Shao
Department of Mechanical Engineering,
The University of Auckland,
Private Bag 92019, Auckland 1142, New Zealand

Chirpreet Singh
Department of Computer Science,
The University of Auckland,
Private Bag 92019, Auckland 1142, New Zealand

Xun Xu*
Department of Mechanical Engineering,
The University of Auckland,
Private Bag 92019, Auckland 1142, New Zealand
Email: xun.xu@auckland.ac.nz
*Corresponding author

Xinfeng Ye
Department of Computer Science,
The University of Auckland,
Private Bag 92019, Auckland 1142, New Zealand

Abstract: Cloud manufacturing is a model for enabling on-demand network access to a shared pool of reconfigurable manufacturing resources that can be rapidly encapsulated, provisioned and released as manufacturing services. Enterprises involved in cloud manufacturing networks share their heterogeneous business models, manufacturing resources and knowledge to provide high quality consumable manufacturing services. Therefore, there is a need to develop a resource description protocol and service description language to cover all the aspects of cloud-based business collaborations in manufacturing industry. This information model needs to be inclusive of representation of all phases of a product’s lifecycle, service model description, and other essential information contributed to e-businesses. This paper presents an ontology-based approach to enable semantic interoperability throughout the whole process of service provision in the clouds. The detailed requirements for enabling cloud-based data exchange are discussed. A generic ontology development process is then proposed with special focus on reusing existing international and/or industrial standards. Specifically highlighted is a systematic guidance on developing ontologies. The utilisation of the proposed ontology in resource virtualisation and resource retrieval in cloud manufacturing environments is also elaborated.

[Received 14 January 2014; Revised 24 April 2014; Accepted 17 May 2014]
Keywords: ontology; cloud manufacturing; STEP/STEP-NC; manufacturing service; manufacturing resource virtualisation.

Reference to this paper should be made as follows: Lu, Y., Shao, Q., Singh, C., Xu, X. and Ye, X. (2014) ‘Ontology for manufacturing resources in a cloud environment’, Int. J. Manufacturing Research, Vol. 9, No. 4, pp.448–469.

Biographical notes: Yuqian Lu is currently a PhD student at the University of Auckland, New Zealand. He received his Bachelor of Engineering with Honours from Dalian University of Technology, China in 2012. His research interest is computer-integrated manufacturing systems, especially cloud manufacturing. He is developing an open manufacturing cloud to encourage effective business interactions between enterprises.

Qun Shao received her Bachelor of Engineering from Zhejiang Sic-Tech University, China, in 2012, and she is currently pursuing her Master of Engineering degree in the Intelligent and Interoperable Manufacturing Systems (IIMS) research group, University of Auckland, New Zealand. Her research focuses on modelling manufacturing resources in cloud manufacturing using ontology.

Chirpreet Singh is a recent graduate from the University of Auckland, New Zealand. He completed his Bachelor of Science (Honours) in Computer Science at the end of 2013. He is currently employed by The University of Auckland as an Information Security Analyst. His plans for the next few years include continuing with his job so that he can gain the vital industry experience that is a requirement for security professionals, before pursuing a PhD in Computer Science.

Xun Xu has a PhD degree from the University of Manchester and has been working in the field of intelligent manufacturing technologies for some 30 years. He is an internationally recognised expert in smart machining systems and more recently cloud manufacturing. He sits on the editorial boards of a number of elite international journals and has published over 200 research papers. He is a Fellow of American Society for Mechanical Engineers, and the Fellow of Institute of Professional Engineers New Zealand.

Xinfeng Ye gained his PhD in Computer Science from the University of Manchester, England, in 1991. He is currently a Senior Lecturer in the Department of Computer Science at The University of Auckland, New Zealand. His current research interests include service computing and cloud computing.

1 Introduction

Cloud is changing the way enterprises do their businesses in that dynamically scalable and virtualised resources are provided as consumable services over the internet (Xu, 2012). Consumers can request services ranging from product design, manufacturing, testing, management and all other stages of a product life cycle (Lu et al., 2013). Service providers can come together to form a temporary alliance to take manufacturing jobs. This process requires distributed manufacturing resources to be encapsulated and packaged as manufacturing services. As a result, geographically isolated manufacturers
integrate and share their heterogeneous business models, manufacturing resources and knowledge to provide configurable solutions. To facilitate effective business interactions in a cloud environment, distributed resources, manufacturing processes, and business models need to be virtualised, encapsulated, and then executed according to customised service requests. Therefore, there is a need to develop resource description protocols and service description languages to cover all the aspects of cloud-based business collaborations in the manufacturing industry. This information model is inclusive of representation of all phases of a product’s lifecycle, service model description, and other essential information contributed to e-commerce.

In terms of information and knowledge sharing, the most straightforward approach is to use a neutral data format for sharing heterogeneous information from multiple enterprises. The credibility of International Organisation for Standardisation (ISO) standards makes it a natural choice for representing information within a certain domain. A typical example of using ISO standard as a neutral data format for data exchange is that of standard for the exchange of product data (STEP) (ISO, 1994) to represent product data through the life cycle of a product. The product information models in STEP are specified in EXPRESS, a modelling language that combines ideas from the entity-attribute-relationship family of modelling languages with object-oriented modelling concepts. The technical objective of STEP is to enable the communication of product data between heterogeneous systems. It provides the basis for representing such data in all phases of the product’s lifecycle. Some of the STEP protocols have been implemented in most of the computer-aided technologies (CAx) and product data management (PDM) systems.

Although STEP provides the communication mechanism for data exchange between heterogeneous systems, the lack of a formal semantic model for EXPRESS schemas and the complexity of EXPRESS itself impose challenges on the extensibility and transportability of data exchange. What is more universal is that, in several instances, concept terms and definitions defined in standards remain largely textual in nature (Gunendran et al., 2007; Young et al., 2007). Moreover, it is sometimes implicit to tell the consistency between certain concepts which are supposed to be interchangeable across standards (Usman et al., 2010). These issues imply that, although standards can ensure somehow common interpretation in relatively confined domains, they sometimes fail to satisfy information interoperability requirement in larger or across domains (Chungoora et al., 2013).

An emerging solution for enabling information exchange is to employ ‘semantic web’ technology, an extension to the existing World Wide Web. In the semantic web, data appearing in the web resources are linked to corresponding entries in ontologies where terms are defined and their relationships are clarified. It provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. This feature can help meet the data sharing requirement in cloud manufacturing – across-company and cross-domain data integration.

In terms of developing an ontology for cloud manufacturing environment, there has been some work dedicated to describing basic properties, capabilities of manufacturing resources (Wang et al., 2012; Yang and Guo, 2012; Zhao et al., 2012; Zhu et al., 2012). Ontology covering manufacturing resources, manufacturing processes, enterprises,
manufacturing requirements and manufacturing capabilities has been partially proposed in these publications. However, there is no well-defined ontology that is able to represent concepts in cloud-based manufacturing businesses, nor a feasible approach to consolidating current industrial ISO standards towards cloud manufacturing ontology.

This paper presents an ontology-based approach to enabling semantic interoperability throughout the whole process of service provision in the cloud. This paper is organised as follows. We discuss the data integration requirements for cloud manufacturing and review two main approaches to data integration in Section 2. A generic ontology development process is presented in Section 3. The ontology for manufacturing resources in cloud environment is proposed in Section 4. In Section 5, we present two examples of utilising the proposed ontology, i.e., resource virtualisation and resource retrieval in a cloud manufacturing system. Finally, discussions and conclusions are given in Section 6.

2 Related research

Cloud manufacturing is a new manufacturing paradigm that aims to transform production-centric business to service-oriented business model. A significant benefit of this business model is that it allows geographically isolated manufacturers to collaborate in large projects via the sharing of virtualised resources. To facilitate business interactions between consumers and service providers, various information needs to be collected and integrated in a cloud manufacturing system (Figure 1). These data inputs include:

- service request information, product details, and service feedback from consumers
- business models, resource information, service information and service logs from manufacturers.

To achieve the seamless information fusion between different data inputs, a unified data format and knowledge structure are required. This data format provides interoperability for data integration in this heterogeneous data environment. By conforming the consolidated data format, chaotic data from various sources are logically interpreted and connected, which enables effective reasoning over the internet and hence a meaningful decision-making process.

There is limited work on data integration in cloud manufacturing systems as cloud manufacturing is a nascent manufacturing paradigm. However, some work has been carried out to enhance interoperability in the distributed manufacturing environment. This effect mainly lays in two aspects:

1. using STEP/STEP-NC as an intermediate for data exchange in manufacturing systems

2. and representing domain knowledge by modelling ontologies for the manufacturing domain.
2.1 *STEP-based data integration*

STEP/STEP-NC is an ISO standard for describing the entire product data throughout the life cycle of a product. It contains different application protocol (AP) to provide data models for targeted applications, activities or environments. In order to achieve the integration through the CAD/CAM/CNC chain, a system was proposed based on STEP data models (Valilai and Houshmand, 2013). In addition to product development centric data integration based on STEP, PDM systems are proposed to share product information via STEP standards. A STEP-compliant PDM system was developed to meet the demand for logically integrated product data, which are physically stocked in a distributed environment (Yang et al., 2009). In this system, a STEP-based PDM schema was proposed to represent and exchange product data among heterogeneous PDM systems. Realising the necessity of exchanging product data over the internet, Makris et al. (2008) developed an approach to providing data integration over the internet via extensible markup language (XML) file. This research combines the STEP standard into XML format, allowing data integration among decentralised business partners in order to facilitate information representation in a cloud manufacturing environment. Wang and Xu (2013) extend STEP by adding definition for service orders, manufacturing services and manufacturing organisations.

Although STEP provides the communication mechanism for data exchange between heterogeneous systems, the lack of semantics and the complexity of EXPRESS itself impose challenges on the extensibility and applicability of data exchange. To be specific, these textural-based ISO standards are undesirable in a cloud-based environment as:

1. Manufacturing business in a cloud environment requires organisations to share product data although the product data from these companies tend to be represented in different data semantics. Therefore, semantic expression is a necessity for data sharing in cloud manufacturing paradigm, whereas ISO standards are unable to consistently express semantic information of product data.

2. Although these ISO standards include organisational data provided by relationships between enterprises or between components of a single enterprise for the purpose of supplier identification, they exclude business planning data such as profit projections, cash flow, and any other organisational data that are essential for online businesses in a cloud environment.
Ontology for manufacturing resources in a cloud environment


2.2 Ontology-based data integration

An emerging solution for information exchange on a broader scale is to employ ‘semantic web’ technology, an extension to the existing World Wide Web. The semantic web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries (W3C Semantic Web Activity, http://www.w3.org/2001/sw/). In semantic web, web documents are properly linked to corresponding terms in a shared vocabulary where concepts of a domain are defined and their relationships are clarified. This vocabulary can also be referred to as ontology. Ontologies formally represent knowledge as a set of concepts within a domain, using a shared vocabulary to denote the types, properties and interrelationships of those concepts.

In the context of cloud manufacturing, there is little work on developing a shared ontology for business interactions between consumers and manufacturers since cloud manufacturing is a relatively new manufacturing model. However, there have been some reports on enhancing data interoperability using ontological approaches within the manufacturing domain. Kim et al. (2006) proposed an ontology-based framework for assembly information sharing for collaborative manufacturing, in which assembly specifications and constraints are represented using ontological constructs. Similarly, Jeong et al. (2010) introduced an ontology-based approach to store human simulation knowledge for validating product design, assembly, and manufacturing operations. In their approach, an ontology is employed as a knowledge base to stock and reuse assembly simulation knowledge for new product variations and revisions.

In addition to ontological data integration throughout the product development chain, ontologies have also been used to enhance business collaborations in manufacturing industry. Ontology-based architecture can facilitate collaboration between different organisations in a manufacturing value chain (Lee et al., 2009), whereby each enterprise can have semantic interoperability across each other for collaborative works using a product ontology. Moreover, Lin et al. (2004) proposed a manufacturing system engineering (MSE) ontology to provide a common understanding of manufacturing-related terms, and therefore to enhance the semantic interoperability and reuse of knowledge resources within global extended manufacturing teams. An ontology-based approach for representing information semantics with a focus on supporting information autonomy that allows the individual team members to keep their own preferred information models rather than requiring them all to adopt standardised terminology is proposed in Lin and Harding (2007).

From existing research outcomes, we can see that an ontological approach for resource virtualisation is able to provide a rigorous mechanism for cross-company and cross-domain data integration in a cloud manufacturing environment as it enriches semantics, reusability and interoperability. Concepts in the existing ISO standards can be sourced as the fundamental inputs for the proposed ontology though these standards may not be directly applicable to cloud manufacturing scenarios. The next section describes the process of ontology development by utilising existing ISO standards.
3 Ontology development process

Ontology development process is an evolutionary design process consisting of proposing, implementing and refining classes and properties that comprise an ontology (Noy and McGuinness, 2001). In terms of ontology development, Noy and McGuinness summarise the development process into seven steps, including: determining the domain and scope of the ontology, considering reusing existing ontologies, enumerating important terms in the ontology, defining the classes and the class hierarchy, defining properties of classes (slots), defining the facets of the slots, and creating instances.

Given the fact that a number of industrial standards are already in use, developing an ontology for manufacturing business is a process of consolidating disparate concepts in textual standards. The applicability of the proposed ontology heavily relies on the quality of the integration process. An ontology enumerating distributed concepts in the manufacturing domain and precisely expressing their relations adds additional value as it allows organisations to maintain on-going practices whilst conforming the proposed ontology. A detailed process of developing the ontology for the manufacturing domain based on existing ontologies and industrial standards is illustrated as follows.

3.1 Step 1: determine the domain and scope of the ontology

The domain of ontology encompasses the information and relations that need to be modelled using the ontology. Undesirable complexity derived from inaccurate scoping can cause low efficiency in the process of utilising the ontology. Therefore, there is a need to clarify the domain and scope of the proposed ontology before investigating the specifications of a proposed ontology. For example, an ontology for PDM throughout the product lifecycle includes all the concepts related to product development from conceptual design to disposal. In our case, an ontology for manufacturing resources is developed for explicitly representing all information and relationships associated with manufacturing process and manufacturing devices.

3.2 Step 2: reuse existing ontologies and standards

Existing ontologies recommend by World Wide Web Consortium (W3C) within the proposed domain, especially those have been acknowledged by the corresponding sectors make it easier to identify key concepts in the suggested domain. Besides, the classes and their relations could be identified through discussions with domain experts and referred to existing industrial standards. Because of the multitude of standards that have been established to support production engineering, it becomes important to incorporate production-centric standards for the proposed domain ontology (Table 1). As is shown in Figure 2, there is overlapping information defined across various standards that contributes to manufacturing resources. Therefore, scoping classes for the proposed ontology involves the reuse of standards in a consolidated way across multiple production-centric standards. In particular, defined concepts in these standards need to be extracted so that the proposed ontology achieves compatibility with existing industry practice.
Table 1  ISO standards and corresponding scope

<table>
<thead>
<tr>
<th>Name</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO 10303</td>
<td>Product data representation and exchange</td>
</tr>
<tr>
<td>ISO 10303-224</td>
<td>Product data representation and exchange – Part 224: Application protocol:</td>
</tr>
<tr>
<td></td>
<td>Mechanical product definition for process planning using machining features</td>
</tr>
<tr>
<td>ISO 10303-238</td>
<td>Product data representation and exchange – Part 238: Application protocol:</td>
</tr>
<tr>
<td></td>
<td>Application interpreted model for computerised numerical controllers</td>
</tr>
<tr>
<td>ISO 10303-239</td>
<td>Product data representation and exchange – Part 239: Application protocol:</td>
</tr>
<tr>
<td></td>
<td>Product life cycle support</td>
</tr>
<tr>
<td>ISO 13399</td>
<td>Cutting tool data representation and exchange</td>
</tr>
<tr>
<td>ISO 13584</td>
<td>Parts library</td>
</tr>
<tr>
<td>ISO 14649</td>
<td>Physical device control – Data model for computerised numerical controllers</td>
</tr>
<tr>
<td>ISO 15531</td>
<td>Industrial manufacturing management data</td>
</tr>
<tr>
<td>ISO 15531-44</td>
<td>Industrial manufacturing management data – Part 44: Information modelling</td>
</tr>
<tr>
<td></td>
<td>for shop floor data acquisition</td>
</tr>
<tr>
<td>ISO 18629</td>
<td>Process specification language</td>
</tr>
<tr>
<td>ISO 9000</td>
<td>Quality management systems – Fundamentals and vocabulary</td>
</tr>
</tbody>
</table>

Figure 2  A manufacturing information organisation perspective for consolidated product information

Source: Chungoora et al. (2013)
3.3 Step 3: define terms and relations among them

The meaning of important terms identified in ISO standards has followed textual definitions that can be varied. The definitions of a term in different standards may emphasise different aspects of the concept based on the target domain and application of a standard. Extensive integration between various definitions for important terms within the target domain are supposed to be carried out, in order to enhance the interoperability of the proposed ontology. This process involves expertise from domain experts and feedback from potential application scenarios. For example, the definitions of a face milling cutter in ISO 14649 – Part 111 (ISO, 2010) and ISO 13399-308 (ISO, 2012) have different emphasis as shown in Table 2. The objective of ISO 13399-308 is to provide a mechanism that is capable of describing product data regarding cutting tools, independent from any particular system, which mainly deals with geometry properties of a given cutting tool. This is the point of view of the ‘service provider’s’. However, ISO 14649 – Part 111 focuses more on functional descriptions of a tool for milling machines. That is, in ISO 14649 – Part 111, key parameters (i.e., dimension, number_of_teeth, hand_of_cut, coolant_through_tool and pilot_length) are defined from the perspective of application concerns. This is therefore the point of view of the user’s. Each attribute contributes to the detailed machining plan for a given part on a milling machine. In 13399-308, the combination of nominated attributes virtualise the detailed geometry of a face milling cutter. In cloud manufacturing, the specifications in both standards are vital to virtualising a face milling cutter, thus calling for a consolidated representation scheme.

Table 2 Definitions of face milling cutter in ISO 14649 and ISO 13399

<table>
<thead>
<tr>
<th>Specifications</th>
<th>ISO 14649 – Part 111</th>
<th>ISO/DTS 13399-308</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension: tool_dimension; number_of_teeth: OPTIONAL INTEGER; hand_of_cut: OPTIONAL hand; coolant_through_tool: OPTIONAL BOOLEAN; pilot_length: OPTIONAL length_measure;</td>
<td>Depth of cut maximum Body diameter Cutting diameter Counterbore diameter connection bore Shank diameter Rake angle axial Rake angle radial Tool cutting edge angle Counterbore depth connection bore Overall length Face effective cutting edge count Face mounted insert count</td>
<td></td>
</tr>
</tbody>
</table>

After identifying important terms in cloud manufacturing, the next step is to ascertain the relations among these terms. In ontology, classes are arranged in a taxonomic hierarchy. There are several possible approaches in developing a class hierarchy (i.e., top-down process, bottom-up process and their combination) (Uschold and Gruninger, 1996). None
of these three methods is inherently better than any of the others. The approach to take depends on the personal view of the domain. Another aspect of assigning the relations among classes is to define the properties of classes. Details to identifying relevant properties will not be investigated extensively as it is not the focus in this research. Further information on generic guidance for ontology development can be referred to in (Noy and McGuinness, 2001).

3.4 Step 4: create instances

The last step is to create individual instances of classes in the hierarchy. Defining an individual instance of a class requires:

1. choosing a class
2. creating an individual instance of that class
3. filling in the slot values.

This process could be viewed as the most straightforward way of virtualising manufacturing resources in cloud manufacturing. Once an instance of a class is created, a piece of virtual resource is generated and mapped to its counterpart in the shop floor.

4 Developing an ontology for cloud manufacturing

The previous section elaborates on the generic process of developing an ontology in the manufacturing domain. This section describes the process of creating an ontology for cloud manufacturing following the above instructions.

Cloud manufacturing responds to customised needs from consumers through collaboratively encapsulating distributed resources and packaging them as services. These resources can be categorised as two types, namely, soft resources and hard resources as shown in Figure 3. Soft resources include software, knowledge, standards, employees, etc. Hard resources refer to facilities that are critical to the production process. Manufacturing capability reflects the ability of a business in undertaking a certain task. This capability includes design capability, experimentation capability, production capability, and management capability.

After the scope is identified, existing ontologies and industry standards are considered. At the time of this research, there is no officially recommended ontology in manufacturing domain on W3C website. In the manufacturing domain, the most notable industry standard is STEP/STEP-NC. STEP-NC is a new generation standard that utilises the application of STEP methods to support computer numerical control machine tools. The standard consists of several parts, each of which is focused on a particular application domain. Some of the parts are listed in Table 3. As mentioned earlier, manufacturing services integrate all kinds of facilities necessary for product development as shown in Figure 3. It should be realised that there is a need to incorporate all the parts in Table 3.

In this case, we adopt an approach that is slightly different from the generic process introduced in Section 3. A Protégé plugin called OntoSTEP (Barbou et al., 2012) allows EXPRESS schemas and Part 21 files to be translated to ontologies represented in the web ontology language (OWL) language, which enhances semantic interoperability by
allowing logical reasoning and inference mechanisms. Hence, we create the proposed ontology by first merging all relevant ISO 14649 parts and then converting them to ontology using OntoSTEP. Following this, we examine the integrity of the ontology and manually add additional compulsory classes and individuals in Protégé, an open-source ontology editor. To be specific, a combined schema of ISO 14649 – Part 10, Part 11, Part 12, Part 111, Part 121 and Part 201 is created and the consistency check is carried out by the online EXPRESS Schema Checker (http://www.steptools.com/translate/expfront) released by STEP Tools, Inc. Then the verified schema is fed into Protégé and converted into an ontology using OntoSTEP. As can be seen from Figure 4, EXPRESS entities and instances are mapped respectively to OWL classes and individuals. Attributes correspond to OWL properties – ObjectProperties link classes together, while DataProperties link classes to data types. A complete summary on the proposed OWL mapping of the basic concepts of EXPRESS is available at (Barbau et al., 2012). It is important to note that OntoSTEP chooses OWL 2 rule language (RL) to represent the ontology for improving the translation performance, which results in the data loss for some restrictions declared in EXPRESS schema. These restrictions defined in EXPRESS schemas include the minimum cardinalities on aggregations, the ENUMERATION type and SELECT type.

Figure 3  Classification of manufacturing resources (see online version for colours)

Table 3  ISO 14649 documentations

<table>
<thead>
<tr>
<th>Part no.</th>
<th>Title</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO 14649: 1</td>
<td>Overview and fundamental principles</td>
<td>2003</td>
</tr>
<tr>
<td>ISO 14649: 10</td>
<td>General process data</td>
<td>2003</td>
</tr>
<tr>
<td>ISO 14649: 11</td>
<td>Process data for milling</td>
<td>2003</td>
</tr>
<tr>
<td>ISO 14649: 12</td>
<td>Process data for turning</td>
<td>2005</td>
</tr>
<tr>
<td>ISO 14649: 111</td>
<td>Tools for milling</td>
<td>2004</td>
</tr>
<tr>
<td>ISO 14649: 121</td>
<td>Tools for turning</td>
<td>2005</td>
</tr>
<tr>
<td>ISO 14649: 201</td>
<td>Machine tool data for cutting process</td>
<td>2011</td>
</tr>
</tbody>
</table>
Next, it is necessary to verify the created ontology against potential application scenarios in cloud manufacturing. The main task is to examine whether manufacturing resources can be readily represented by the created ontology. New classes and properties are considered if essential information of a resource cannot be handled by the ontology. For instance, the generated ontology lacks geometrical description of face milling cutters. In order to enrich its modelling capability, definitions from ISO 13399 are added to the ontology for comprehensive representation of face milling cutters. In some cases, industrial practice represents a more efficient approach on PDM as the declared information for a manufacturing resource makes more sense for market interests. Hence, mature practice from the industry needs to be integrated into the ontology. To demonstrate the integration process, descriptions for cutters from SANDVIK (http://www.sandvik.com/) and descriptions for machine centres from OKUMA (http://www.okuma.com/) are integrated into the ontology after some further simplification of eliminating parameters that contribute less to decision-making in a cloud manufacturing environment. As can be seen from Figure 5, definitions for cutters, such as depth_of_cut_maximum, body_diameter and overall_length from ISO 13399, are imported into the ontology. The ontology is further enriched by integrating selected descriptive declaration for cutters based on industrial practice, especially from machine
tool vendors. It is important to note that this ontology is only created for initialisation in cloud manufacturing systems. An ontology is dynamically evolving as more explicit knowledge is captured from domain experts, consumer complaints and system faults, thus enhancing the system intelligence and decision-making efficiency in return.

Figure 5  Further integration on machine centres and cutters (see online version for colours)
5 Utilising the ontology

Ontology provides a solid mechanism to represent real world objects and perform inferencing tasks with the help of a software reasoner. This section discusses how it facilitates manufacturing resource virtualisation and resource retrieval.

5.1 Virtualising manufacturing resources

Until now, ontology has been initialised by incorporating international standards and industrial practice. In general, it is a good idea to make sure the ontology has been well-structured before inserting extensive numbers of instances. Changes to class or slot structure after instances have been entered may cause information loss; inconsistencies may arise, too. In addition, adding slots to classes will result in filling in the slot values for all instances that were created previously. Therefore, the ontology needs to be verified by domain experts and a considerable number of use cases. In terms of virtualising manufacturing resources on top of the ontology, there are two approaches:

1. creating individuals in the ontology editor, Protégé Editor for example
2. developing a semantic web application based on APIs for ontology to manipulate the ontology.

This section will elaborate the implementation process for both approaches.

Protégé allows users to instantiate real world objects through a graphical user interface (GUI). After creating an instance, entering values for corresponding slots, and specifying necessary relationships with other existing instances, a new manufacturing resource is virtualised and stored into the ontology. In an ontology, instances and classes are declared in the same file, which is different from that in STEP. For instance, Figure 6 shows a virtualised face milling cutter – RA210-038C3-09M delivered by SANDVIK. It is denoted as an individual of face_milling_cutter and all the relevant assertions are created, thus the description of this particular cutter is archived in a neutral fashion, independent from any particular system. Figure 7 is a screen shot of the graphical representation of a small subset of the resulting ontology. It showcases three different OWL individuals each of which represents a tool instance. Furthermore, the screen shot depicts a typical representation of the tool hierarchy from the resulting ontology. It shows that a twist_drill is a subclass of drill which itself is a subclass of milling_machine_tool_body, which itself is a subclass of tool_body.

The alternative approach is to develop an independent application based on available APIs to load, create and modify ontologies. In this case, the OWL API (http://owlapi.sourceforge.net/) is selected to communicate with the ontology. The OWL API is a free, open-source Java library for managing OWL ontologies. The API provides the ability to load and save ontologies, to query ontologies, and to perform inferencing tasks over ontologies with the use of reasoners. A prototype is implemented using Java (JDK version 7 update 25); specifically the JavaFX platform (version 2.2.25). Figure 8 shows how the application is used to insert new cutters into the ontology directly through the interface. The new tool instances are inserted by asserting axioms representing those tools into the runtime representation of the ontology.
After inserting a new instance of a particular tool from the detailed specification by a manufacturer, a new OWLNamedIndividual object is instantiated (terms in Arial Narrow font are interfaces defined in OWL API). Each parameter specified by the user, for example, model/name, requires a new OWLDataPropertyAssertionAxiom object to model the value before it is finally asserted into the ontology through the OWLOntologyManager object. As the classes into which these new tool instances are inserted have been specified explicitly in the implementation, there is no risk of the resulting ontology becoming inconsistent. However, it should be noted that in the current prototype, there is no automated validation done for the values specified by the user for new tools. This implies that although the resulting ontology will be virtually guaranteed to be consistent, the data properties asserted from the values specified by the user could result in invalid OWL individuals (invalid tool instances).
5.2 Retrieving manufacturing resources

Reasoning OWL ontologies is another advanced function that ontologies can offer for manipulating structural data. This feature enables users to search for appropriate manufacturing resources based on customised searching rules, which connects service request with satisfying manufacturing resources.

A reasoner is a key component for working with OWL ontologies, especially for retrieving individuals from ontologies. In fact, virtually all querying of an OWL ontology (and its imports closure) should be done using a reasoner. This is because knowledge in an ontology might not be explicit and a reasoner is required to deduce implicit knowledge so that correct query results are obtained. The OWL API includes various interfaces for accessing OWL reasoners. Reasoners, such as FaCT++, Hermit and Pellet, provide implementations of the OWL API OWLReasoner interface.

The choice of software reasoner involves some consideration for the extensibility of the prototype. There are additional efforts to enrich knowledge capabilities of OWL ontologies, one of which is the introduction of the semantic web rule language (SWRL). SWRL (http://www.w3.org/Submission/SWRL/) is an expressive OWL-based rule language based on subset of the RuleML language that adds more capabilities to OWL by introducing rules into ontologies. It is considered as an extension to OWL as it permits rule-based inference over OWL ontologies. It provides users the possibility of modelling
the implicit knowledge in the manufacturing domain that cannot be represented by OWL into ontologies. This feature will largely improve the modelling capability of ontology-based knowledge base. Therefore, a reasoner that supports SWRL rules is desired, although SWRL rules do not contribute major functionality to the current version of the prototype. The Hermit reasoner has limited support for SWRL (at the time of this paper). For instance, it does not allow for SWRL built-in. FaCT++ does not support SWRL rules at all. For these reasons, and especially due to the inclusion of the Pellet reasoner within the latest OWL-API distribution package, Pellet is chosen as the reasoning software for the prototype.

In the following example (Figure 9), we show detailed steps to retrieve a manufacturing resource. In this example, a machining request in STEP-NC format is loaded into the prototype and the working steps are extracted by the prototype automatically. From there, description logic (DL) queries are automatically generated from the user’s input. The selection of DL queries is because Pellet would only interpret description logic safe rules (DL-safe rules), which means they are more relevant and efficient alternatives to SWRL for the prototype.

Figure 9  Cutter retrieval for a given machining plan (see online version for colours)

A DL query is effectively a set of OWL class expressions. An OWL class expression is an ontological construct that is used to represent sets of instances’ properties. That is, if an instance of a particular class satisfies a set of property conditions, it is considered an instance of the respective class expressions that match it. DL queries are specified using the Manchester OWL syntax, and consist of class expressions. For instance, Figure 10
depicts the set of class expressions that can be used to compose a DL query. It highlights that a DL query can consist of any number of combinations form the listed constructs, and the level of expressiveness afforded by class expressions.

**Figure 10** Various constructs for a DL query (see online version for colours)

```
Class Expression (DL query):
Class | ObjectIntersectionOf | ObjectUnionOf |
ObjectComplementOf | ObjectOneOf | ObjectSomeValuesFrom |
ObjectAllValuesFrom | ObjectHasValue | ObjectHasSelf |
ObjectExactCardinality | DataSomeValuesFrom |
DataAllValuesFrom | DataHasValue | DataMinCardinality |
DataMaxCardinality | DataExactCardinality
```

Figure 11 shows a typical DL query that is generated by our prototype. It selects a tapered end mill with respective conditions for the five parameters, i.e. diameter, edge_radius, number_of_teeth, tool_offset_length and overall_assembly_length. This query is forwarded to the PelletReasoner object to realise. This process is initiated in the DLQueryFormatter class that is responsible for gathering the input DL queries and formatting the DL query responses for the prototype’s graphical interface. From there, a DLQueryEngine object is instantiated through which the actual class expressions are parsed. Finally, each DL query is sent to the PelletReasoner object through the getInstances method to infer all the matching tools from the associated ontology, using its internal tableaux-based (Sirin et al., 2007) reasoning algorithm. The PelletReasoner object then returns a NodeSet object containing all the matching tool instances, each represented by its own OWLNamedIndividual object. These results can be further retrieved and displayed to users through GUI as displayed in Figure 9.

**Figure 11** An example DL query for retrieving manufacturing resource (see online version for colours)

```
DL Query (Class Expression) :
(tool_body) and
(tapered_endmill) and
(diameter some decimal [<= "20.0"^^decimal]) and
(edge_radius some decimal [<= "1.5"^^decimal]) and
(number_of_teeth value "4.0"^^decimal) and
(tool_offset_length some decimal [<= "80.0"^^decimal]) and
(overall_assembly_length some decimal [<= "80.0"^^decimal])
```

This section provided a detailed discussion on how semantic web technologies facilitate resource virtualisation and retrieval in cloud manufacturing environment. Manufacturing resources can either be inserted into the ontology through an ontology editor or a standalone application via OWL API. Resource retrieval can be achieved by incorporating a software reasoner that supports rule-based inference over OWL ontologies.
6 Discussions and conclusions

In this paper, we presented an approach to enable resource virtualisation and resource retrieval in a cloud manufacturing environment. This mainly deals with the development of the ontology for cloud manufacturing and inferencing over the ontology. The generic process of developing the ontology for cloud manufacturing entails:

1. determining the domain and scope of the ontology
2. reusing existing ontologies/standards
3. defining terms and relations among them
4. creating instances (resource virtualisation).

Following the suggested process, an ontology for cloud manufacturing resources was developed and it is believed to be the first of its kind in manufacturing domain. The usage of existing standards gives the proposed ontology with additional credibility and reusability, as well as allows it to be better integrated with existing legacy PDM systems. In addition, we developed a prototype to showcase the process of resource virtualisation and resource retrieval by working with the OWL-API and the Pellet software reasoner.

Following the proposed process, one is able to develop an ontology for a target domain or application activity. The most crucial of all for ontology creation is the proper consideration of existing ontologies and standards. This is particularly critical for cloud manufacturing ontology as it calls for large-scale integration of heterogeneous knowledge from enterprise architecture to the shop floor. It is also imperative to note that ontology development is not a static process. Instead, ontologies evolve as knowledge and technology advance. Every small aspect of manufacturing businesses generates new information continuously. This information may include: dynamic service requests from consumers, new business intelligence from enterprises, novel business processes, advanced manufacturing resources, and even unexpected system faults at shop floor. New knowledge needs to be extracted from these knowledge sources and embedded into the ontology so that it is able to output meaningful reasoning results. This requires an additional module that is able to communicate with the proposed ontology by manipulating the ontology based on new knowledge gathered from business practices. From the ontology end, communication with ontologies can be achieved by utilising OWL-API as it supports powerful capabilities of manipulating ontologies. Meanwhile, further integration can be carried out by using some APIs of legacy knowledge gathering systems.

Resource retrieval deals with querying ontologies using dedicated query languages in the back-end. DL queries are used in this research to obtain a particular class that satisfies a set of property conditions. The key to accurate retrieval is to create high quality queries with precise refinement conditions. These refinement conditions are domain knowledge gathered from engineering practices. In this research, we append domain knowledge to DL queries, whereas it can be represented as a collection of rules and embedded into ontology itself, which simplifies query constructs. SWRL rules are a perfect choice for describing truth and rules within a target domain, and advanced reasoners, such as Pellet, supports reasoning over rule-based ontologies. This feature brings more feasibility for complex reasoning in manufacturing applications as product development process
involves heavy *if-then* judgments based on practical experience. Additionally, rule-based ontologies can be agilely expanded as domain knowledge evolves by simply modifying the embedded rule sets.

In terms of deploying resource virtualisation and resource retrieval module in cloud manufacturing systems, this module can be packaged as consumable web services so that any web-based cloud manufacturing system is able to access and invoke its main functions to support smart decision-making. Web services extend the World Wide Web infrastructure to provide the means for software to connect to other software applications. Applications access web services via ubiquitous web protocols and data formats such as HTTP, XML, and simple object access protocol (SOAP), with no need to worry about how each web service is implemented. This feature overcomes the barrier between different operating systems and programming languages, thus improving interoperability between different applications. Standalone service package for resource virtualisation and resource retrieval in cloud environment makes a significant reduction in system development efforts and maintenance costs. It generates more substantial benefits as distributed cloud manufacturing systems can utilise an integrated knowledge base from domain experts, thus enabling knowledge sharing in a large-scale.

The OWL ontologies generated by the proposed approach have several applications beyond the examples described in this paper. One potential application is to facilitate complex decision-making processes in cloud manufacturing environment. The objective of this research at University of Auckland is to develop an AI-based expert system to enable effective decision-making requirements in cloud manufacturing businesses. The current strategy for service composition and encapsulation in cloud manufacturing is to use numerical simulation methods. This approach seems to be less effective in real cases due to the existence of great number of expertise from engineering practices. We believe ontology-based approach better facilitates decision-making on resource retrieval, service composition, and service delivery.

References


