International Journal of Production Research
Vol. 47, No. 23, 1 December 2009, 6525–6542

Taylor & Francis
Taylor & Francis Group

Managing distributed manufacturing knowledge through multi-perspective modelling for semantic web applications

W.Y. Zhang[a], M. Cai[b]*, J. Qiu[c] and J.W. Yin[b]

[a]*School of Information, Zhejiang University of Finance and Economics, Hangzhou, China;* [b]*School of Computer Science, Zhejiang University, Hangzhou, China;* [c]*School of Computer Science, Hangzhou Dianzi University, Hangzhou, China*

(*Received 15 May 2008; final version received 24 June 2008*)

The development and maintenance of semantic web (SW) means that collaborative manufacturing systems are faced with increasing challenges caused by the growing difficulty in managing distributed manufacturing knowledge. This paper presents a multi-perspective modelling approach to systematically manage distributed manufacturing knowledge on the SW. Considering knowledge engineering as a cyclic and constructive modelling process, a multi-perspective knowledge modelling process is proposed to evolve along knowledge elicitation, engineering modelling, UML-based object modelling, OWL-based ontology modelling, knowledge formalisation and OWL-QL assisted knowledge verification activities. The proposed approach is viewed as a promising knowledge management method that facilitates the implementation of computer supported cooperative work (CSCW) in distributed manufacturing for SW applications by integrating the industrial, UML enabled software engineering techniques into recent ontology-based knowledge engineering process. The feasibility of knowledge management through multi-perspective modelling is manifested using the manufacturing ontology for manufacturing electronic connectors.

Keywords: distributed manufacturing; knowledge engineering; knowledge management; ontology modelling; semantic web (SW); unified modelling language (UML); web ontology language (OWL)

1. Introduction

Owing to recent advances in the field of artificial intelligence (AI), knowledge engineering has been successfully applied in many engineering areas, especially in knowledge-aided manufacturing (KAM). Triggered by the trends in the manufacturing field towards highly specialised solution providers cooperatively offering configurable manufacturing services in remote sites and production units, it is becoming an increasing need for collaborative teams to establish and maintain a computer supported cooperative work (CSCW) through semantic inter-operability at the knowledge level. Although the current computer-integrated manufacturing (CIM) and computer network technologies have laid the foundation for the emerging fields of CSCW in distributed manufacturing, the heterogeneity of distributed manufacturing knowledge representation is still a major

---

*Corresponding author. Email: cm@zju.edu.cn

ISSN 0020–7543 print/ISSN 1366–588X online

DOI: 10.1080/00207540802311114
http://www.informaworld.com

obstacle to managing the knowledge for cross-enterprise and cross-disciplinary collaboration.

The recent semantic web (SW) initiative (Berners-Lee *et al.* 2001) tries to establish better semantic connections between different knowledge-based resources to build distributed knowledge management systems more efficiently. Domain ontology is the most prominent part of the SW research. Ontology provides the critical semantic foundation of formalised knowledge representation. As the most expressive semantic mark-up language, the web ontology language (OWL) (McGuinness and Harmelen 2004) facilitates greater content processing by contributing heavily to the wide spread use of ontologies. This language has a well-understood semantic basis but lacks both a wide user community outside AI research laboratories and a standard graphical representation, which are important considerations for aiding the human comprehension of ontologies (Cranefield 2001a).

Recently, unified modelling language (UML) (Rumbaugh *et al.* 1998) was designed for human-to-human communications of diverse models in the whole lifecycle of software engineering. UML has become an expressive and standardised modelling language in industrial software development processes. UML has a standard graphical representation, a huge and rapidly expanding user community, and a high level of commercial tool support (e.g. CASE). As a consequence, the scope of UML is broadening to include more declarative modelling tasks, in particular, object modelling. Some researchers (e.g. Falkovych *et al.* 2003) have recognised the importance of UML for object-oriented conceptual modelling, especially on the SW. The effort of combining UML with OWL will remove one of the most commonly stated criticisms of the suitability of OWL used alone for ontology modelling on the SW, due to the latter's definition in terms of a formal representation language rather than an expressive graphical model.

Aiming at representing distributed manufacturing knowledge explicitly and formally and sharing it between multiple agents in the distributed manufacturing, this paper presents a multi-perspective modelling approach to systematically manage distributed manufacturing knowledge on the SW. Considering knowledge engineering as a cyclic and constructive modelling process, a multi-perspective knowledge modelling process is proposed to evolve along knowledge elicitation, engineering modelling, UML-based object modelling, OWL-based ontology modelling, knowledge formalisation and knowledge verification activities. The UML-based object models serve as a graphical and structured basis for conceptual communication between domain experts and knowledge engineers. The OWL-based ontology models extend the UML-based object models with added semantics using a classification capability designed into the object models themselves, enabling to bridge the gap between the structured representation in the object models and the ability to cope with all this with machine-processable semantics in OWL format. Formal knowledge representation in OWL format extends common engineering modelling with capabilities of knowledge sharing and distributed problem solving to support semantic inter-operability between multiple agents. The ontology query or reasoning support for OWL can be exploited for verifying the completeness and consistency of formal knowledge representation in OWL format.

Based on our experience in exploring SW technologies for ontology-based modelling in collaborative engineering design (Zhang and Yin 2008a, b), the authors show in this paper how to manage distributed manufacturing knowledge more efficiently by integrating the industrial, UML enabled software engineering techniques into ontology-based knowledge engineering process. The feasibility of knowledge management through multi-perspective modelling, especially from UML to OWL transformation, is manifested using the manufacturing ontology for manufacturing electronic connectors. The elaborated manufacturing system, which initially contains informal product-process-resource relationships, has been modelled by the manufacturer to support standalone product design, but not distributed collaboration in e-manufacturing (Molex 1999). The proposed multi-perspective modelling approach in the present paper allows the inherent product–process–resource relationships in the manufacturing ontology to be defined explicitly and formally, so that the semantic access and retrieval of manufacturing components across different enterprises and disciplines becomes possible.

2. Related work

The knowledge management research community has come a long way towards taking a modelling perspective on knowledge engineering. The modelling approach represents an effort to obtain a better understanding, description and representation of the problem. With the modelling approach, development of knowledge management systems can be faster and more efficient through the reuse of existing models for different areas of the same domain. Specifically, the effort at knowledge modelling usually proceeds along mediating representation, task analysis, or ontology modelling, to which the science of knowledge engineering has much to contribute.

The importance of knowledge modelling in knowledge management has been identified in CommonKADS (Valente *et al.* 1998), which supports structured knowledge engineering techniques, provides tools for corporate knowledge management and includes methods that perform a detailed analysis of knowledge intensive tasks and processes. A suite of mediating models including organisation model, task model, agent model, communication model, expertise model and design model form the core of its systematic knowledge management methodology. The MIKE approach (model-based and incremental knowledge engineering) (Angele *et al.* 1998) takes the expertise model of CommonKADS as its general model pattern and provides a smooth transition from a semiformal representation (structure model), to a formal representation, and further to an implementation oriented representation (design model).

On the other hand, research in the growing field of ontology modelling offers a firm basis for solving knowledge modelling problems. The main motivation behind ontology is to establish standard models, taxonomies, vocabularies and domain terminologies, and use them to allow for sharing and reuse of knowledge bodies in computational form. The process specification language (PSL) project (Schlenoff *et al.* 1996) at the American National Institute of Standards and Technology (NIST) develops a unified ontology for representing manufacturing process to serve as an interlingua to support process-related inter-operability throughout the manufacturing lifecycle. The adaptive holonic control architecture (ADACOR) project (Leitao *et al.* 2005) defines a domain-specific proprietary manufacturing ontology, expressed in an object-oriented UML-based manner. A shared terminology relevant to manufacturing processes is provided to model taxonomy of manufacturing components, with which multiple agents can inter-operate effectively to support collaborative production automation and control.

Ontologies are also expected to play a major role on the SW. The emerging SW possesses a huge potential to overcome knowledge modelling difficulties over the web, by modelling the concepts in a knowledge domain with a high degree of granularity and formal structure including references to mutually agreed-on semantic definitions in ontologies. An example of the use of SW in knowledge modelling is configuration knowledge representations (Felfernig *et al*. 2003), which compare the requirements of a general configuration ontology with the logics chosen for the SW, and describe the specific extensions required for the purpose of communicating configuration knowledge between state-of-the-art configurators via OIL and DAML + OIL. Lin and Harding (2007) propose a manufacturing system engineering (MSE) ontology model on the SW for inter-enterprise collaboration. The MSE ontology provides common understanding of manufacturing-related terms via OWL, and therefore enhances the semantic interoperability and reuse of knowledge resources within globally extended manufacturing teams. In our previous work (Zhang and Yin 2008a, b), an ontology-based modelling in collaborative engineering design is developed on the SW, enabling multiple design agents to share a clear and common understanding of the definitions of engineering design problems and the semantics of exchanged engineering design knowledge. Manda *et al*. (2006), Qiu *et al*. (2007), Georgoudakis *et al*. (2007) and Ye *et al*. (2008) also proposed ontology-based modelling on the SW for design, manufacturing or engineering collaborations across ubiquitous virtual enterprises. As SW languages are relatively new languages – having only become official W3C standards since 2001 – their use in the engineering field, in particular, distributed manufacturing area has not yet reached the pervasive level that has been seen in the information technology world.

Notwithstanding the promising results reported from existing research work for model-based knowledge management, the modelling frameworks either lack ontological support or are closely tied to a specific ontology language, especially there has been little research using the incremental ontology-based knowledge engineering approach to support the management of distributed manufacturing knowledge for SW applications. A specific ontology language used alone for ontology modelling lacks both a wide user community and an expressive graphical representation.

Though the use of mediating representation, task analysis and ontology modelling are all important for knowledge modelling, the mediating representation and ontology modelling are essential, since executable knowledge bases are not only organised from the perspective of humans, but also for the convenience of the representation and reasoning mechanisms of the performance environment. Effective mediating representation through ontology modelling may be optimised not only for machine efficacy, but also for human understanding. This paper focuses on the use of mediating representation through ontology modelling, so as to incrementally develop a manufacturing ontology on the SW by integrating the industrial, UML enabled software engineering techniques into ontology-based knowledge engineering process. The seamless integration provides common understanding for a consistent and generic description of distributed manufacturing knowledge shared between multiple agents, facilitating the implementation of CSCW in distributed manufacturing.

3. Knowledge engineering through multi-perspective modelling

Distributed manufacturing is a very complex process, which involves plenty of cross-enterprise and cross-disciplinary manufacturing knowledge throughout the manufacturing lifecycle. The involved knowledge is often located geographically and represented in heterogeneous formats. This makes effective capture, retrieval, reuse, sharing and exchange of knowledge a critical issue in a distributed manufacturing environment. In order to model the distributed manufacturing knowledge in a manner that is explicit, formal, complete, embedded in its context and yet comprehensible, a multi-perspective knowledge modelling approach is required. It is proposed to view knowledge engineering as a cyclic and constructive modelling process rather than a prototyping process. New elicitation may lead to a refinement, modification or decomposition of the already built-up models. On the other hand, the evolving models may guide the further knowledge acquisition. The domain experts and knowledge engineers should collaborate with each other. The proposed knowledge engineering process consists of knowledge elicitation, engineering modelling, object modelling, ontology modelling, knowledge formalisation, and knowledge verification activities (Figure 1).

Knowledge elicitation covers the interactions with manufacturing domain experts through a series of knowledge acquisition sessions in order to elicit manufacturing knowledge of the domain and produce a federated, distributed description of it. Knowledge engineers must explain the objective of the sessions, the process and approach to acquire manufacturing knowledge, and the expected results. Methods such as structured

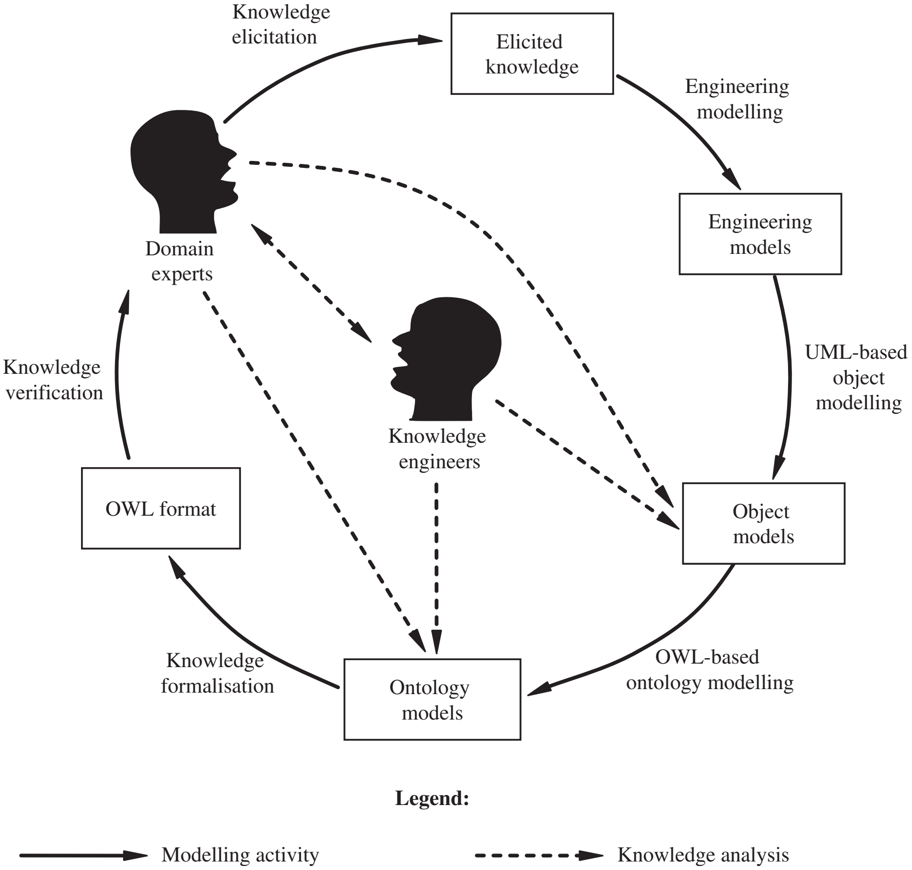

Figure 1. Knowledge engineering through multi-perspective modelling.

interviews, observation and structuring techniques can be used in the knowledge elicitation activity for acquiring informal descriptions of manufacturing knowledge about a specific domain.

The distributed manufacturing process includes various engineering phases such as product design, process planning and resource scheduling, and is traditionally represented as domain-specific engineering models and transitions between them. The initially elicited, informal descriptions of distributed manufacturing knowledge with associated contextual information, are analysed and embodied into these engineering models using different engineering modelling tools and technologies, such as commercial computer-aided design (CAD), computer-aided process planning (CAPP), enterprise resource planning (ERP) and product lifecycle management (PLM) packages.

Though various engineering modelling tools and technologies are able to describe and distinguish involved manufacturing knowledge while maintaining efficiency and computability in standalone, one-off engineering modelling environments, they have limitations. For example, the elements in distributed manufacturing knowledge cannot be defined in isolation, which affects modularity and reusability of the knowledge representation. Therefore, UML-based object modelling is adopted to analyse the engineering models in order to structure them and develop object models that are used as graphical and structured basis for conceptual communication between domain experts and knowledge engineers. UML is used as an aid in structuring and describing the domain-specific manufacturing knowledge independently of any particular implementation.

One limitation of the UML-based object modelling is that it is just a graphical notation for human-to-human communication based on the object-oriented paradigm, but not formal enough to be used at compile time and run time for reasoning about formal models such as ontologies, which therefore prohibits knowledge re-use and automated reasoning in distributed manufacturing. Towards combining engineering models effectively across borders to support distributed collaboration, an ontological description to the distributed manufacturing knowledge is necessary to be exploited in the ontology modelling activity. The importance of ontology as a central building block of the SW has brought a convergent work on the development of manufacturing ontology for SW enabled knowledge modelling in distributed manufacturing. The key concepts of manufacturing knowledge are represented as different inter-related ontologies through OWL-based ontology modelling, which extends the UML-based object models with added semantics using a classification capability designed into the object models themselves. The resulted OWL-based ontology models serve as a common foundation to the unified definition of distributed manufacturing problem and the formal semantics of exchanged manufacturing knowledge, enabling the bridging of the gap between the structured representation in the object models and the ability to cope with all this with machine-processable semantics in OWL format. Formal knowledge representation in OWL format extends traditional engineering modelling with capabilities of knowledge sharing and distributed problem solving to support cooperation between distributed multiple agents.

Finally, the ontology query or reasoning support for OWL can be exploited for verifying the completeness and consistency of formal knowledge representation in OWL format, often through a set of test cases. The OWL-QL (Fikes *et al.* 2003), a formal language for deductive query answering on the SW, precisely specifies the semantic relationships among queries, query answers, and the knowledge bases used to produce the answers, and can be used for semantic search that exploits the domain ontologies, semantic indexes and semantic relationships built in the ontology models. The OWL-QL query-answering dialogues for knowledge verification may be further enhanced by integrating the ontology reasoning capabilities of the description logic (DL) reasoner such as Racer (Haarslev and Moller 2003) that performs run time class subsumption checking, classification of individuals, terminological and assertion reasoning, and so on.

Having realised the advantages of both popular knowledge representation techniques – UML and OWL for knowledge modelling, the proposed approach is viewed as a promising knowledge management method to build manufacturing ontology for distributed manufacturing on the SW.

4. An illustrative example in building manufacturing ontology through multi-perspective modelling

This section elaborates an illustrative example in building manufacturing ontology for manufacturing electronic connectors through multi-perspective modelling. The elaborated manufacturing system, which initially contains informal product–process–resource relationships, has been modelled by the manufacturer to support standalone product design, but not distributed collaboration in e-manufacturing (Molex 1999).

The terminal-housing assembly is a key sub-assembly in connector design. It refers to the method used to join a terminal and a conductor in the plastic housing. Good termination assures sound electrical contact and maximum strength between the conductor and the terminal. Connectors are created through a multi-phase manufacturing process including stamping, assembling, drilling and milling. In assembly, terminals are inserted into housings to make the conductor and insulator one unit. Various assembling methods including terminal stitching and gang insertion can be used, depending on the functional requirements of products.

In this section, the authors focus on UML-based object modelling, OWL-based ontology modelling and OWL-QL assisted knowledge verification for SW applications.

4.1 *UML-based object modelling*

The association of object models with engineering models allows for conceptual communication about the engineering models between domain experts and knowledge engineers. Object models need not describe engineering models completely, only contextual interpretation of the model components needs to be included. The interpreted components are associated with classes in the corresponding domains by means of class instances created in the object-oriented knowledge base. Figure 2 shows a sub-part of UML-based object model, i.e. UML class diagram of manufacturing system for manufacturing electronic connectors, which contains conceptual network of domain-specific classes (e.g. *Manufacturing*, *Product*, *Process* and *Resource*), attributes and relationships (e.g. inheritance, association, aggregation) between them.

The modelling of *Manufacturing* class is fundamental to the description of all tangible or intangible manufacturing objects the authors observe in the distributed manufacturing environments, and the relationships between them. The main types of *Manufacturing* class of interest, i.e. its child classes through inheritance, are *Product*, *Process* and *Resource* classes. The *Product* class contains all its technical and geometrical attributes and describes the structure of a product. The *Process* class contains all its process-related attributes and

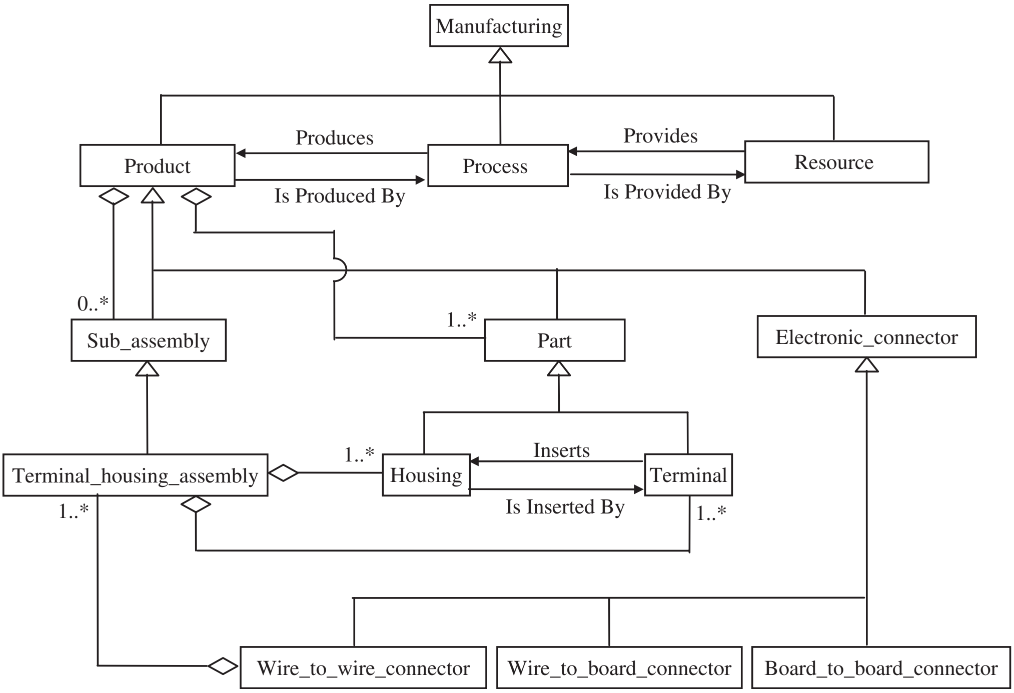

Figure 2. UML-based object modelling to manufacturing class.

describes how to produce a product. The *Resource* class contains all its physical and logical attributes and describes what is required to provide a process. The *Resource* class associates with the *Process* class through *Provides* association. The *Process* class associates with the *Product* class through *Produces* association. Some associations may have inverses. For example, the inverse of *Provides* is *IsProvidedBy*; the inverse of *Produces* is *IsProducedBy*.

4.1.1 *UML-based object modelling to product class*

Referring to Figure 2, the *Product* class is usually specialised by *Sub_assembly* and *Part* classes. Each product may comprise some sub-assemblies and parts. Each sub-assembly may comprise lower-level sub-assemblies and parts.

In the manufacturing field of electronic connectors, the *Product* class is specialised by *Electronic_connector* class, which is further specialised by *Wire_to_wire_connector*, *Wire_to_board_connector* and *Board_to_board_connector* classes, etc. Each of these product classes may comprise some sub-assemblies and parts. For example, the *Wire_to_wire_connector* class comprises a *Terminal_housing_assembly* class that is a sub-class of *Sub_assembly* class. The *Terminal_housing_assembly* class comprises *Housing* and *Terminal* classes that are sub-classes of *Part* class. The *Terminal* class associates with the *Housing* class through *Inserts* association. The inverse of *Inserts* is *IsInsertedBy*.

4.1.2 *UML-based object modelling to process class*

Figure 3 shows a sub-part of UML-based object model of *Process* class. The *Process* class is usually specialised by *Process_plan* and *Operation* classes. Each process plan may

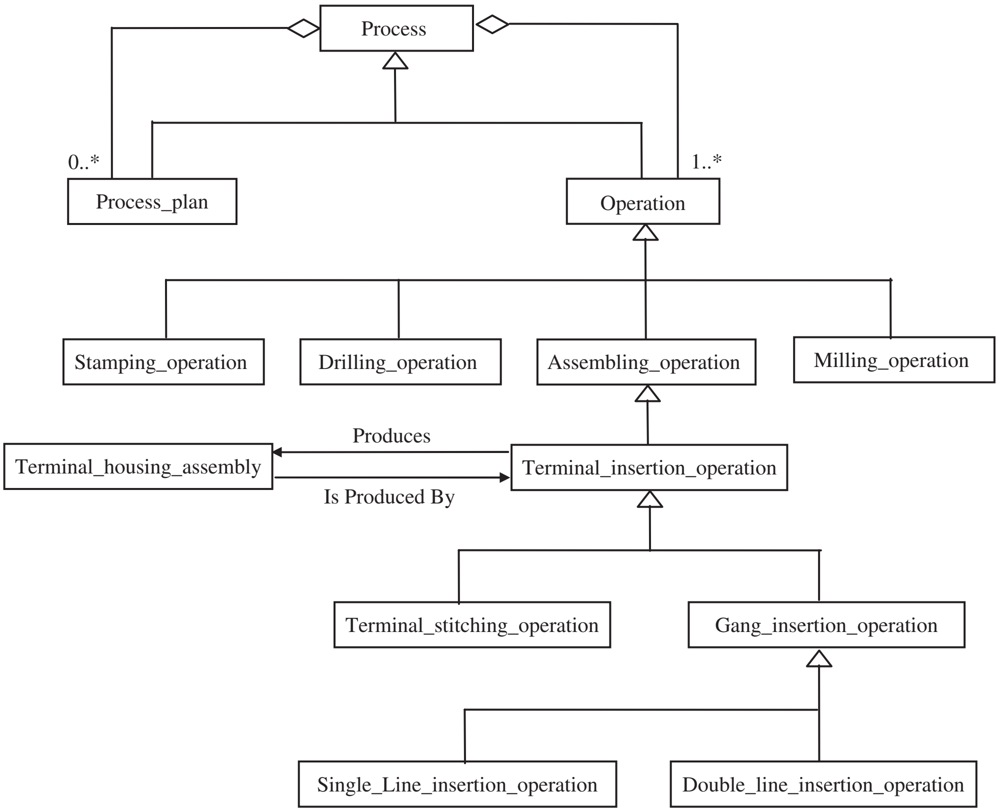

Figure 3. UML-based object modelling to process class.

comprise some sub-process plans and operations. The *Operation* class is usually specialised by *Stamping_operation*, *Assembling_operation*, *Drilling_operation* and *Milling_operation* classes, etc.

In the manufacturing field of electronic connectors, the *Assembling_operation* class is specialised by *Terminal_insertion_operation* class, which is further specialised by *Terminal_stitching_operation* and *Gang_insertion_operation* classes. The *Gang_ insertion_ operation* class is specialised by *Single_line_insertion_operation* and *Double_line_ insertion_operation*. The *Terminal_insertion_operation* class associates with the *Terminal_housing_ assembly* class through *Produces* association. The inverse of *Produces* association is *IsProducedBy*. In other words, given a specific sub-assembly like terminal-housing assembly, users may find its desired operation like terminal insertion operation in the process planning phase, following the *Produces* association between both.

4.1.3 *UML-based object modelling to resource class*

Figure 4 shows a sub-part of UML-based object model of *Resource* class. The *Resource* class is usually specialised by *Equipment*, *Human_resource*, *Shop_floor*, and *Raw_material* classes, etc. The *Equipment* class is usually specialised by *Machine*, *Tool*, *Gripper* and *Robot* classes, etc.

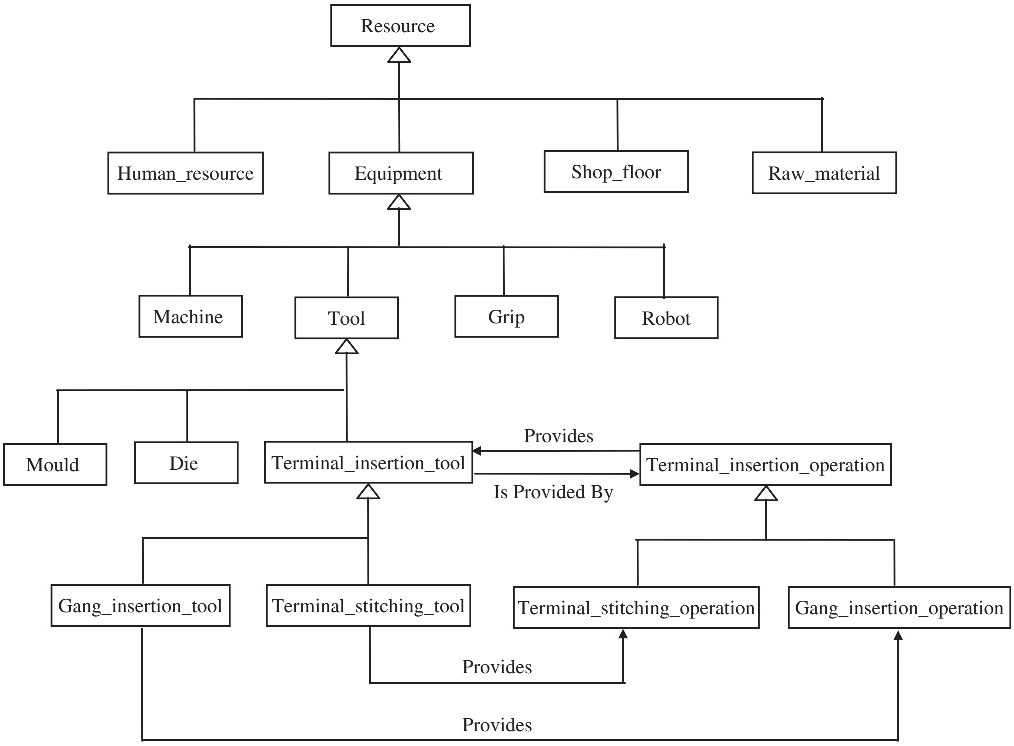

Figure 4. UML-based object modelling to resource class.

In the manufacturing field of electronic connectors, the *Tool* class is specialised by *Mould*, *Die* and *Terminal_insertion_tool* classes, etc. The *Terminal_insertion_tool* class is specialised by *Termainl_stitching_tool* and *Gang_insertion_tool* classes. The *Terminal_insertion_tool* class associates with the *Terminal_insertion_operation* class through *Provides* association. In other words, given a specific operation like terminal insertion operation, users may find its desired equipment like terminal insertion tool in the resource scheduling phase, following the *Provides* association between both. The inverse of *Provides* association is *IsProvidedBy*. Similarly, the *Termainl_stitching_tool* class associates with the *Terminal_stitching_operation* class through *Provides* association; the *Gang_insertion_tool* class associates with the *Gang_insertion_operation* class through *Provides* association.

4.2 *Mapping between the UML-based object model and OWL-based ontology model*

The proposed UML-based object modelling enables the manufacturing knowledge to be organised in a structured manner with a standard graphical representation. However, UML alone for knowledge modelling still lacks a formal representation scheme, which prohibits easy knowledge exchange and automated reasoning over the web at run time. Though the object models are non-executable conceptual models, their association with formal ontology models makes it possible to reason about the object models automatically. The authors use OWL for ontology modelling, which maps and extends

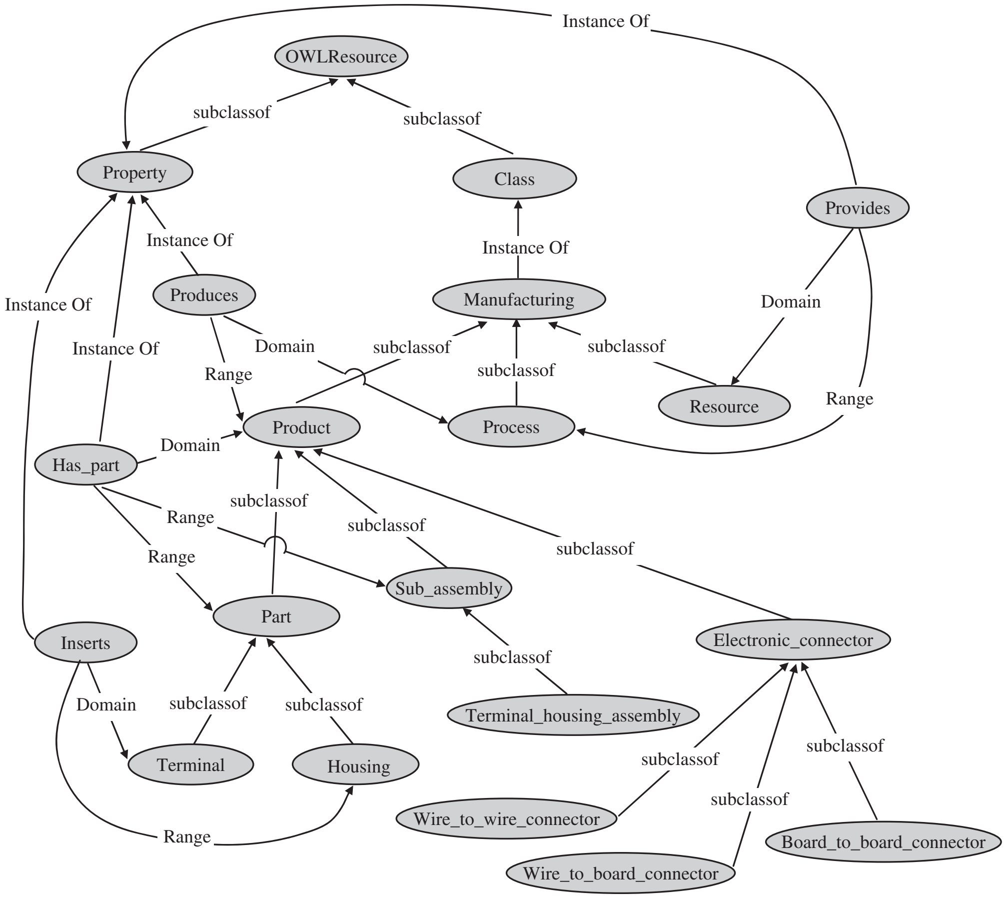

Figure 5. OWL-based ontology modelling to manufacturing class.

the UML-based object models with added semantics by means of a classification capability designed into the object models themselves.

To realise the mapping between UML-based object model and OWL-based ontology model, the authors first serialise UML model in XML Metadata Interchange (XMI) representation (OMG 2002) using a UML tool (e.g. Poseidon for UML – http://www.gentleware.com). Then an eXtensible Stylesheet Language Transformation (XSLT) processor (e.g. Xalan – http://xml.apache.org) (Cranefield 2001b) can be utilised to transform the XMI representation into OWL format and Java classes using the mapping rules, some of which are defined below.

A UML class in the object model can be mapped to a corresponding OWL class in the ontology model. For example, the *Manufacturing* class in the object model (Figure 2) is mapped to the *Manufacturing* class in the developed ontology model (Figure 5).

The 'association' relationship in the object model is mapped to an OWL property with certain domains and ranges in the ontology model. For example, The *Produces* 'Association' relationship between *Process* and *Product* classes in the object model (Figure 2) is mapped to the *Produces* OWL property with *Process* class as domain and

Product as range in the developed ontology model (Figure 5). To make the ontology model more concise and more readable, the inverses (e.g. *IsProducedBy*) of some OWL properties (e.g. *Produces*) are hidden.

The 'aggregation' relationship in the object model is mapped to a *Has_part* OWL property with certain domains and ranges in the ontology model. For example, the 'aggregation' relationship between *Product* and *Sub_assembly* classes in the object model (Figure 2) is mapped to the *Has_part* OWL property with *Product* class as domain and *Sub_assembly* as range in the developed ontology model (Figure 5).

The 'inheritance' relationship in the object model is mapped to the *subClassOf* OWL property in the ontology model, which is defined in the *de facto* W3C standard (McGuinness and Harmelen 2003).

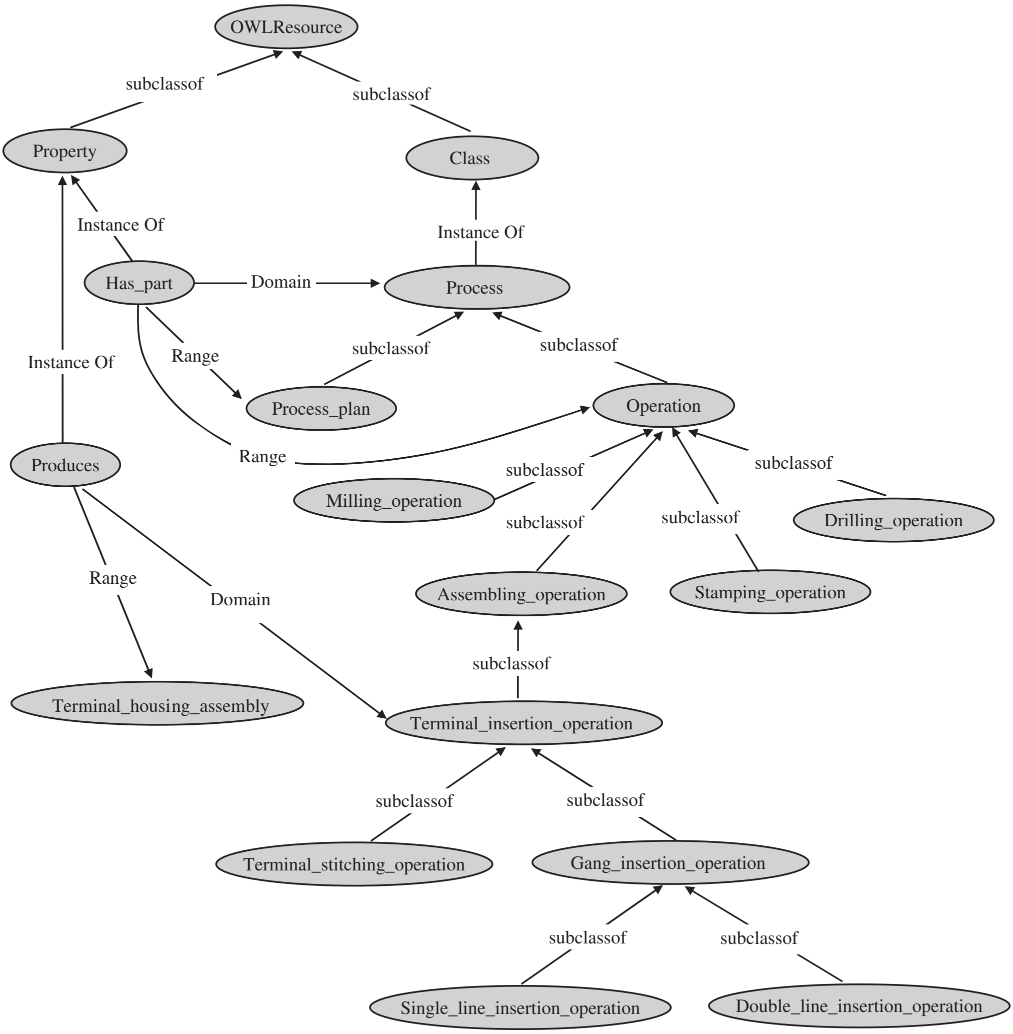

Figure 6. OWL-based ontology modelling to process class.

Similarly, the OWL-based ontology modelling to *Process* class and that to *Resource* class are developed in Figure 6 and Figure 7 respectively.

By means of the above XSLT-based transformation, an OWL document is produced as the output. Figure 8 shows the representative snippets of OWL format of the developed manufacturing ontology corresponding to the combined OWL graph in Figures 5, 6 and 7. It is displayed using Internet Explorer's XML parser. The outputted OWL format can be imported into a widely accepted ontology editor, e.g. Protégé-2000 (Gennari *et al.* 2003) for further refinement.

4.3 *Defining semantic index for the product models*

Once the manufacturing ontology is fully implemented, the user can then add instances of a class in the ontology as individuals. In other words, when the user builds an engineering

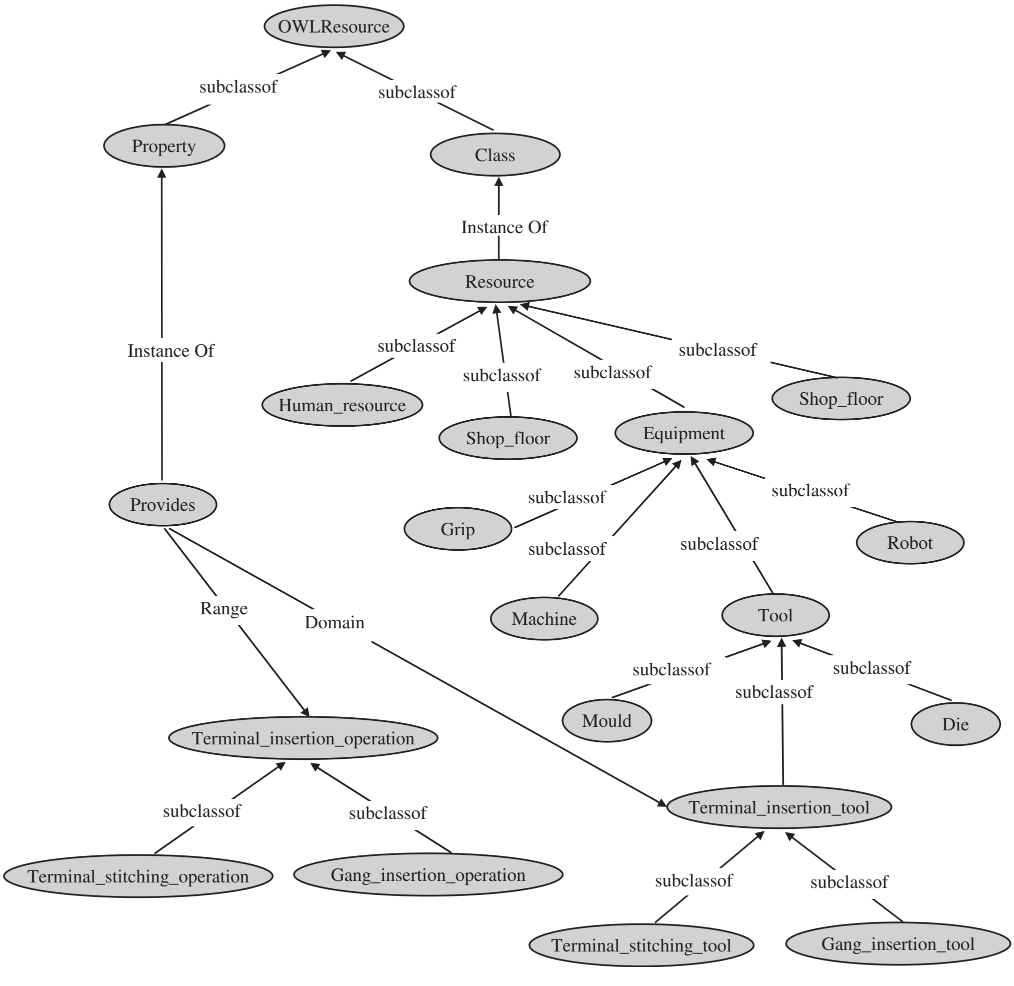

Figure 7. OWL-based ontology modelling to resource class.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/manufacturing-ontology.owl#"
  xml:base="http://www.owl-ontologies.com/manufacturing-ontology.owl">
 ……
 <owl:Class rdf:ID="Terminal_housing_assembly">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty>
     <owl:ObjectProperty rdf:ID="IsProducedBy"/>
    </owl:onProperty>
    <owl:someValuesFrom rdf:resource="#Terminal_insertion_operation"/>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:someValuesFrom rdf:resource="#Terminal"/>
    <owl:onProperty>
     <owl:ObjectProperty rdf:ID="Has_part"/>
    </owl:onProperty>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:someValuesFrom rdf:resource="#Housing"/>
    <owl:onProperty>
     <owl:ObjectProperty rdf:about="#Has_part"/>
    </owl:onProperty>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Sub_assembly"/>
 </owl:Class>
 ……
 <owl:ObjectProperty rdf:ID="Produces">
  <owl:inverseOf>
   <owl:ObjectProperty rdf:about="#IsProducedBy"/>
  </owl:inverseOf>
  <rdfs:domain rdf:resource="#Process"/>
  <rdfs:range rdf:resource="#Product"/>
 </owl:ObjectProperty>
 ……
 <Terminal_insertion_operation rdf:ID="Terminal_insertion_operation05">
  <Produces>
   <Terminal_housing_assembly rdf:ID="Terminal_housing_assembly01">
    <Has_part>
     <Housing rdf:ID="Housing06"/>
    </Has_part>
    <IsProducedBy rdf:resource="#Terminal_insertion_operation05"/>
    <Has_part>
     <Terminal rdf:ID="Terminal03"/>
    </Has_part>
   </Terminal_housing_assembly>
  </Produces>
 </Terminal_insertion_operation>
 ……
</rdf:RDF>
```

Figure 8. Sample of OWL source codes of the developed sub-part of manufacturing ontology.

model of any of the domain world, e.g. by using a CAD tool, he/she can associate it with a corresponding class in the UML object diagram, which is further mapped to the OWL class in the ontology, i.e. creating a semantic index. Though the engineering models are non-executable informal models, their association with formal ontology models via graphical object models makes it possible to reason about the engineering models automatically.

An example of the semantic annotation of the CAD engineering model of a *Terminal_housing_assembly* is shown in Figure 9. The CAD engineering model is mapped to the intermediate UML object diagram that is further mapped to the final OWL instance model of Protégé ontology editor, enabling to create semantic indexes for the illustrated sub-assembly *Terminal_housing_assembly01* (an instance of *Terminal_housing_assembly* class). The OWL instance model shows that *Terminal_housing_assembly*01 has parts *Terminal03* (an instance of *Terminal* class) and *Housing06* (an instance of *Housing* class), and is produced by *Terminal_insertion_operation05* (an instance of *Terminal_ insertion_ operation* class).

4.4 *OWL-QL assisted knowledge verification*

Ontology modelling to the manufacturing knowledge has provided a semantic network of domain concepts, intertwined with diverse relationships and property taxonomies. Corresponding to OWL used for formal knowledge representation, OWL-QL is employed to verify the completeness and consistency of developed manufacturing knowledge base in OWL format. For example, Figure 10 shows an OWL-QL query example 'If the last run finds a spoiled *Process*205 whose type is *Terminal_stitching_operation*, then show me a *Process* that *Produces Terminal_housing_assembly*, and I don't like *Terminal_stitching_operation*'.

The semantic index to associate the process with the ontology model has been created, the *Double_line_insertion_operation* is a kind of *Process* that *Produces Terminal_housing_assembly*, and the *Double_line_insertion_operation* is not a kind of *Terminal_stitching_operation*, therefore an instance of *Double_line_insertion_operation* will be retrieved.

5. Conclusions

As the SW shapes the future of the Web, the SW enabled knowledge management is playing a more and more important role in enterprise application development. However, the exploration of SW technologies for distributed management of manufacturing knowledge has been impeded by a gap between emerging ontology engineering tools and industrial software engineering tools. This paper describes a systematic approach to bridging this gap through multi-perspective modelling – that is, the representation, sharing and exchange of manufacturing knowledge from different viewpoints including engineering modelling, object modelling and ontology modelling. The UML-based object models serve as a graphical and structured basis for conceptual communication between domain experts and knowledge engineers. The OWL-based ontology models serve as a common foundation to the unified definition of distributed manufacturing problem and the formal semantics of exchanged manufacturing knowledge, enabling to bridge the gap between the object models and final formalised representation in OWL format. The evolution of UML

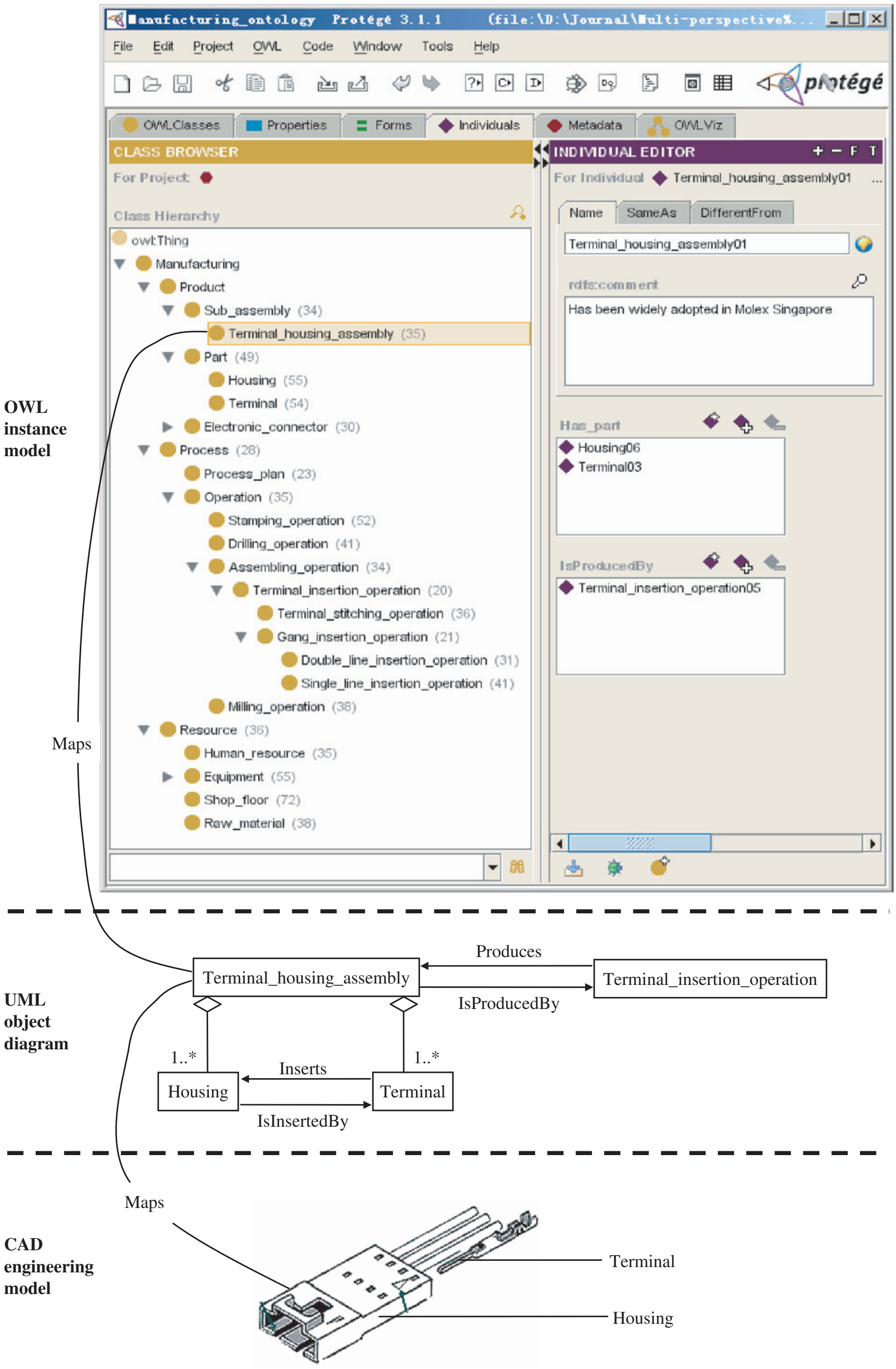

Figure 9. Defining semantic index to associate engineering models with ontology models via intermediate object models.

```
Premise: (type Process205 Terminal_stitching_operation)
Query Pattern: {(Produces?pTerminal_housing_assembly) (type ?p Process) (¬(type ?p
Terminal_stitching_operation))}
Must-Bind Variables List: (?p)
May-Bind Variables List: ( )
Don't-Bind Variables List: ( )
Answer KB Pattern:{ http://www.owl-ontologies.com/manufacturing-ontology.owl}
```

Figure 10. An OWL-QL query example in verifying ontology models.

towards full use of the SW has been implemented in XSLT with the aid of mapping rules. The ontology query or reasoning support for OWL is exploited for verifying the completeness and consistency of formal knowledge representation in OWL format. Combined knowledge representation of UML and OWL brings ontology development process closer to wide practitioner's population and allows for automated reasoning about the object models.

Although the OWL-based ontology models may reduce the flexibility of knowledge representation in comparison with UML-based object models, they enable to avoid *ad hoc* modelling and to obtain consistent, standardised, sharable and exchangeable knowledge models. This can lead to significant improvement in searching, browsing, integration and reasoning of heterogeneous Web services for manufacturing collaboration, along with improved reusability of distributed manufacturing knowledge.

Our future work in knowledge engineering will look into developing and publishing more manufacturing ontologies in an integrated UML and OWL representation using the proposed approach, in order to capture an extensive set of vocabularies of general manufacturing with a community-wide agreement. A more thorough and robust mapping between UML and OWL will also be elaborated in the follow-up work.

Acknowledgements

The work has been supported by the National High-Tech R&D Program, China (No. 2006AA01Z170, No. 2007AA01Z124), the National Natural Science Foundation, China (No. 60703042), and the Zhejiang Natural Science Foundation of China (No. Y106045, No. Y1080170).

References

Angele, J., *et al.*, 1998. Developing knowledge-based systems with MIKE. *Automated Software Engineering*, 5 (4), 398–418.

Berners-Lee, T., Hendler, J., and Lassila, O., 2001. The semantic web. *Scientific American*, 284 (5), 34–43.

Cranefield, S., 2001a. Networked knowledge representation and exchange using UML and RDF. *Journal of Digital Information*, 1 (8), Article No. 44. Available from: http://jodi.tamu.edu/Articles/v01/Cranfield/ [Accessed 15 February 2001].

Cranefield, S., 2001b. UML and the Semantic Web. *Proceedings of the international semantic web working symposium*, 30 July–1 August 2001, Stanford University, Stanford, CA. Stanford, CA: Stanford University, 113–130.

Falkovych, K., Sabou, M., and Stuckenschmidt, H., 2003. UML for the Semantic Web: transformation-based approaches, *In*: B. Omelayenko and M. Klein, eds. *Knowledge transformation for the semantic web*. Amsterdam, Netherlands: IOS Press, 92–106.

Felfernig, A., *et al.*, 2003. Configuration knowledge representations for SW applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: Aiedam*, 17 (1), 31–50.

Fikes, R., Hayes, P., and Horrocks, I., 2003. *OWL-QL – A language for deductive query answering on the semantic web*. Stanford, CA: Knowledge Systems Laboratory, Stanford University.

Gennari, J.H., *et al.*, 2003. The evolution of Protégé-2000: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58 (1), 89–123.

Georgoudakis, M., *et al.*, 2007. Methodology for the efficient distribution a manufacturing ontology to a multiagent system utilising a relevant Meta-Ontology. *Proceedings of IEEE international conference on emerging technologies & factory automation*, 25–28 September 2007, Patras, Greece. Patras, Greece: University of Patras, 1210–1260.

Haarslev, V. and Moller, R., (2003). Racer: a core inference engine for the Semantic Web. *Proceedings of the 2nd international workshop on evaluation of ontology-based tools*, 20 October 2003, Sanibel Island, Florida. NY, IEEE Computer Society, 27–36.

Leitao, P., Colombo, A., and Restivo, F., 2005. ADACOR: a collaborative production automation and control architecture. *IEEE Intelligent Systems*, 20 (1), 58–66.

Lin, H.K. and Harding, J.A., 2007. A manufacturing system engineering ontology model on the Semantic Web for inter-enterprise collaboration. *Computers in Industry*, 58 (5), 428–437.

Manda, J., *et al.*, 2006. A methodology for product family ontology development using formal concept analysis and Web Ontology Language. *Journal of Computing and Information Science in Engineering*, 6 (2), 103–113.

McGuinness, D.L. and Harmelen, F.V., 20 March 2004. OWL web ontology language overview [online]. Available from: http://www.w3.org/TR/2004/REC-owl-features-20040210/ [Accessed 20 March 2008].

Molex Electronics Ltd., 1999. Basic connector engineering – an introduction to connectors and connector technology [online]. Available from: http://www.molex.com/training/bce/index.html [Accessed 20 March 2008].

OMG XMI Specification, 2002. OMG document format/02–01–01 [online]. Available from: http://www.omg.org/cgi–bin/doc?formal/2002–01–01 [Accessed 20 March 2008].

Qiu, H.B., *et al.*, 2007. An agent- and service-based collaborative design architecture under a dynamic integration environment. *International Journal of Advanced Manufacturing Technology*, 35 (1), 15–25.

Rumbaugh, J., Jacobson, I., and Booch, G., 1998. *The unified modelling language reference manual*. Reading, MA: Addison-Wesley.

Schlenoff, C., Knutilla, A., and Ray, S., 1996. Unified process specification language: Requirements for modelling process. Interagency Report 5910. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA.

Valente, A., Breuker, J., and de Velde, W.V., 1998. The *Common* KADS library in perspective. *International Journal of Human-Computer Studies*, 49 (4), 391–416.

Ye, Y., *et al.*, 2008. An ontology-based architecture for implementing semantic integration of supply chain management. *International Journal of Computer Integrated Manufacturing*, 21 (1), 1–18.

Zhang, W.Y. and Yin, J.W., 2008a. Exploring SW technologies for ontology-based modelling in collaborative engineering design. *International Journal of Advanced Manufacturing Technolog*, 36 (5), 833–843.

Zhang, W.Y. and Yin, J.W., 2008b. Weaving a Semantic Grid for multidisciplinary collaborative design. *International Journal of Production Research*. Published online first. DOI 10.1080/00207540701769966.