



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

BY: **Attribution.** You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<https://creativecommons.org/licenses/by-nc-nd/2.5/>

An inter-enterprise semantic web system to support information autonomy and conflict moderation

H K Lin¹, J A Harding^{2*}, and P C Teoh³

¹Department of Industrial Engineering and Management, I-Shou University, Taiwan, Republic of China

²Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, UK

³Motorola Technology Malaysia, Penang, Malaysia

The manuscript was received on 21 February 2005 and was accepted after revision for publication on 18 July 2005.

DOI: 10.1243/095440505X32788

Abstract: This paper discusses a semantic web architecture for formation of an extended project team manufacturing system engineering moderator (EEMSEM) which includes four major modules: ontology acquisition, ontology mapping, knowledge acquisition, and design moderation. This collaborative system architecture focuses on how to support information autonomy that allows individual enterprises to keep their own preferred terminology or languages rather than requiring them to adopt a single standardized vocabulary. Different engineering information terminologies are interpreted and automatically connected to the corresponding terminologies through mapping into the mediated ontology model. A case study is provided to demonstrate how the EEMSEM applies its ontology during the moderation of an extended enterprise, supply chain focused project.

Keywords: semantic web, ontology, inter-enterprise collaboration, information autonomy, conflict moderation

1 INTRODUCTION

The global webs of a supply chain, with increasing dependence on multidisciplinary, multi-enterprises, and multinational contributors, offer alternative tactics to gain competitive advantages, to exploit market opportunities, and to outsource external competencies as they occur. Globally distributed inter-enterprise teamwork facilitates manufacturing engineering agility through collaboration in which information is shared seamlessly with partners to create the right design in a shorter time frame. The current web technology, such as internet, intranet, and extranet, has provided platform independence for users to publish and access data anywhere and at any time to support global network collaboration. However, the consequent enormous amounts of various heterogeneous data (e.g. semantic heterogeneity or structural heterogeneity) make it

increasingly difficult to share and exchange information required by a wide variety of users. In response to this problem and to achieve true information autonomy and interoperability, the concept of 'semantic web' [1, 2], i.e. machine-processable semantics of data on the web, start to emerge and become a reality.

Apart from information issues connected with the semantic web technology, the major goal of most manufacturing organizations is the development and adoption of global inter-enterprise operational approaches that require companies to coordinate their activities effectively and efficiently across normal enterprise boundaries. It is a very complex task to support concurrent manufacturing operation and improve cooperation and coordination activities within any extended project team in an extended enterprise or virtual enterprise environment. The concepts of a manufacturing system engineering (MSE) moderator to support a MSE team have been suggested and previously reported [3, 4]. This concept has been adopted and combined with the application of semantic web technology to expand its operational scope to extended project teams, thus providing an extended project teams

*Corresponding author: Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Ashby Road, Loughborough LE11 3TU, UK.
email: j.a.harding@lboro.ac.uk

manufacturing system engineering moderator (EEMSEM).

2 CONCEPT OF THE EEMSEM

The main function of the EEMSEM is to coordinate expertise and support the role of concurrency within the engineering activities of extended project teamwork. The growing complexities of engineered systems are generally performed by multiple enterprises contributing to multidiscipline project teams. The design or redesign of any part of a manufacturing system must satisfy many different requirements and objectives, so compromises generally have to be made to achieve a balanced design for the new or re-engineered manufacturing system. Members of project teams participating in any aspect of manufacturing system design or re-engineering must therefore be aware (or be made aware) when decisions they are taking may have a significant effect on other team members. For example, considered decisions may constrain or even compromise other contributions to the re-engineering process. When teams are small and can meet regularly to discuss the project, team members are easily made aware of the requirements and views of other people. However, when teams are large and located at multiple sites (or different global locations) this awareness can be difficult to achieve, and the task is further complicated when team members come from an extended project team environment.

Typically, people working within a particular company or group will develop their own vocabulary, or common terms for particular issues, elements, or activities that they work with frequently. Even when contributors work in the same domain, these different team members may have totally different backgrounds and use different terminologies. Therefore, different MSE information models may be used by different parts of the extended manufacturing project teams. This results in the occurrence of two common types of problem [5] in communication: first, the same term is applied to different concepts (semantic problem); and second, different terms may be used to denote the same entity. This may be owing to errors in the use of terminology (syntax problem) or because parallel terminology has been developed by different parties independently, resulting in further semantic problems.

Therefore, the major goals of the EEMSEM are:

1. Provide an interoperability mechanism with well-defined semantic definitions of an MSE ontology model [6, 7], which is committed to by all participating extended project team partners. The model allows each of the partners to keep his or

her own individual language via mapping to the cross-understanding MSE ontology to support information autonomy.

2. Reduce the complexity of EEMSE systems by providing a set of knowledge of the profiles and characteristics of participants within the extended project team group and communication mechanisms to orchestrate dialogues between them. The communications mechanisms are used to disseminate information about detected 'conflict' or potential 'conflict' (if a design change made by designer A has implications or causes problems for designer B, it is said to cause conflict).

3 ARCHITECTURE OF THE EEMSEM AND FUNCTIONAL MODULES

The proposed design of the EEMSEM is as an intelligent software system operating on an extranet-based platform which is open and supports execution of distributed 'MSE agents'. The term 'MSE agents' was used to refer to each combination of engineer(s) and supporting web application software(s) performing an identifiable function to contribute to the developing MSE design. For example, the project management function may be fulfilled by a project agent, which may be expected to include software tools to support both strategic management and project planning. The developed approach enables plug-in of the EEMSEM to any extended project team's extranet platform directly, as shown in Fig. 1.

The EEMSEM includes four major modules: ontology acquisition, ontology mapping, knowledge acquisition, and design moderation. The designs of the knowledge acquisition module (KAM) and the design moderation module (DMM) are largely from the implementation of MSE moderator in the MISSION project as described in the references [4, 8]. The main contribution to the EEMSEM in this paper lies in the application of a new ontology approach and semantic web technology for knowledge and information integration. For this purpose, an ontology acquisition module (OAM) and the ontology mapping module (OMM) provide a language interoperability service and allow the individual enterprise to keep its own individual language through mapping on to the mediated MSE ontology model. The details of each module are discussed in the following sections.

3.1 Ontology acquisition module

The proposed design of the OAM is to establish a common, mediated, or integrated MSE ontology which allows MSE users to access various heterogeneous data repositories through mapping on to the MSE ontology. Since different MSE information

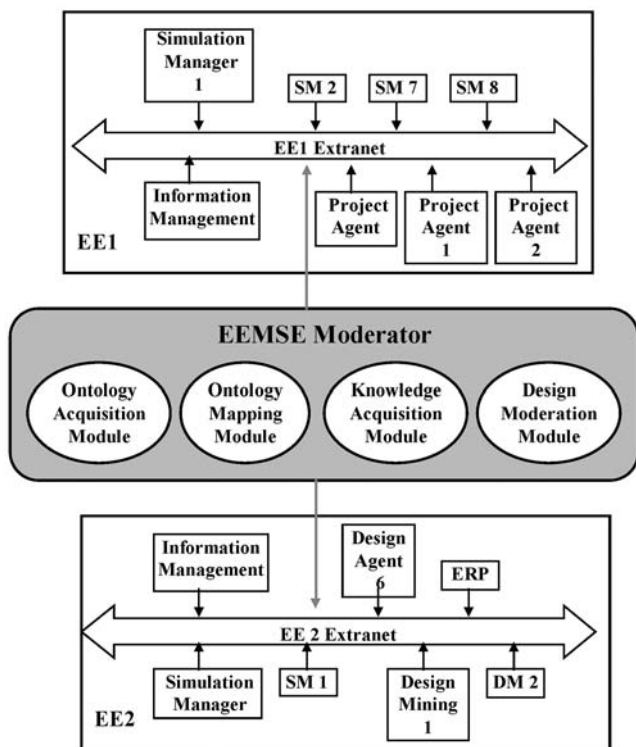


Fig. 1 The general architecture for the EEMSEM (EE1 = extended project team 1; EE2 = extended project team 2)

models will have been independently developed by different enterprises or MSE design agents, they will include semantic heterogeneity (e.g. different vocabularies, logical schemas) and structural heterogeneity (e.g. different data structures: plain files, databases, and internet documents). One of the first steps in developing the OAM is to acquire the common ontology created by a particular extended project team group, describing explicit knowledge in a well-defined terminology that is accepted by all participating engineers, and this is called the extended project team (EE) ontology. The EE ontology needs to be built to meet the needs and objectives of the particular interdisciplinary project. Additionally, the EE ontology should be extensible so that it can be changed as necessary when the structure of an extended project team in the extended enterprises and virtual enterprises environment is changed. Extended enterprises and virtual enterprises are generally disbanded as soon as their goals have been achieved, i.e. when their project is completed, and the participating companies will go their individual ways, or recombine to form further extended enterprises or virtual enterprises to address new opportunities and challenges [9].

In the proposed architecture, the MSE ontology model (as described in reference [7]) is used to illustrate the manufacturing system domain and cover all the terminology aspects and needs for an

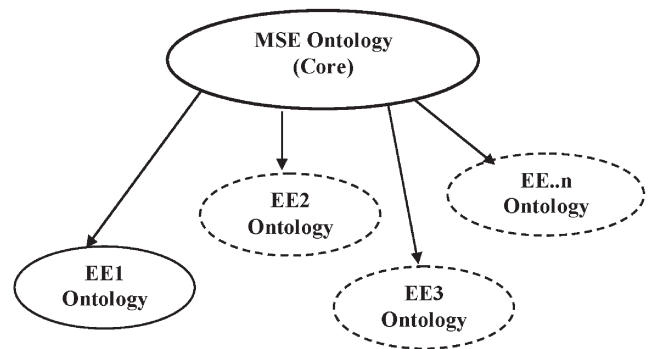


Fig. 2 EE ontology architecture

EEMSEM. It therefore serves as a core for the complete, extensible, or re-organized structure of the individual EE ontology, shown in Fig. 2.

The complete definition of the EE ontology, mapping information, and mapped target ontology are stored and accessed through the EE ontology server. Figure 3 shows the instantiation of the general architecture for the EE1 or EE2 or ... EE**n* ontology server of OAM on the EE1 or EE2 or ... EE**n* extranet platform.

The EE ontology server provides the mediated terminology for the individual enterprise's documents within this particular EE group and therefore each enterprise can use its own individual language through mapping into the mediated EE ontology. In addition, the EE ontology server stores the information about the mapped target ontology. Therefore, individual MSE design agents from different enterprises could share information and exchange documents through the EE ontology server. That is, the proposed design of the EEMSEM enables it to see and interpret the information stored in the EE ontology server and use the content to perform its moderation activities. Mapping details of any identified 'change' into the neutral EE ontology enables the EEMSEM to perform most of its moderation activities by using its own, single chosen language. The mapping is carried out by the OMM, and the functionality and structure of this module are discussed in the next section.

3.2 Ontology mapping module

Ontology mapping is the process by which two ontologies are semantically related at the conceptual level, thereby transforming instances from the source ontology into instances in the target ontology according to those semantic relations [10, 11] (shown in Fig. 4). In this research, all the individual ontologies must be mapped to the mediating ontology that specifies the shared semantics of the concepts. Two steps have been identified and embedded into the OMM (displayed in Fig. 4) of the EEMSEM: normalization and ontology mapping rules.

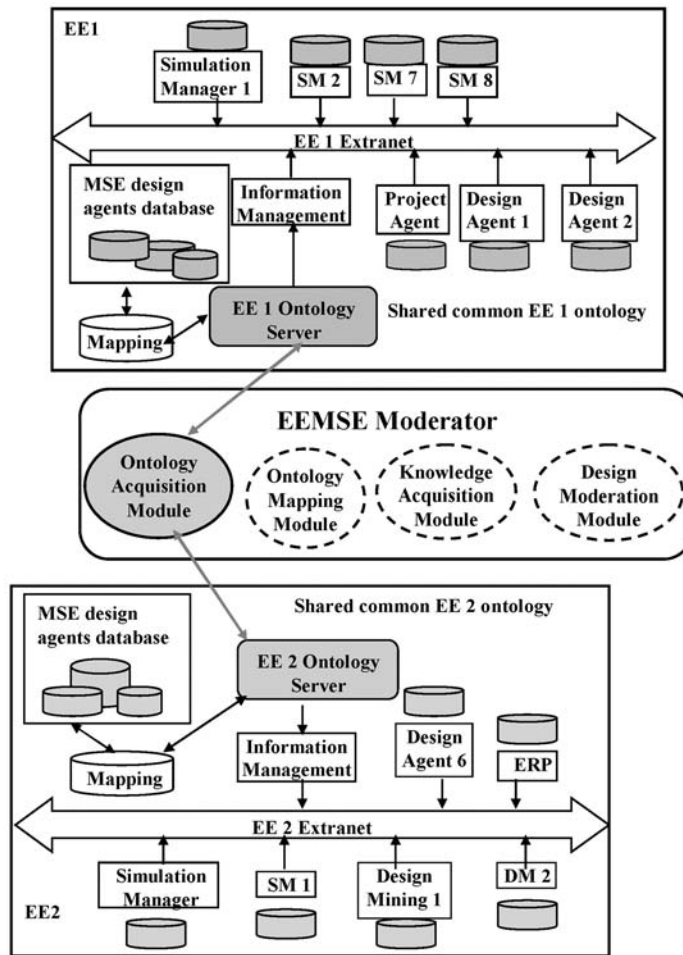


Fig. 3 EE ontology server of the OAM in the EEMSEM

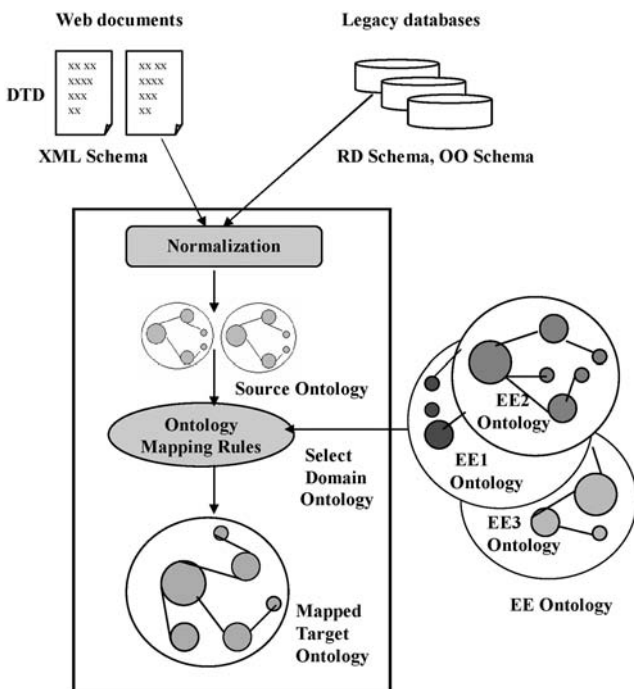


Fig. 4 Architecture for OMM

3.2.1 Normalization

Maedche [10] pointed out that normalization extends the ontology-mapping problem somewhat to the problem of integrating existing information sources that are not ontology based. For example, in most industries, there are large quantities of existing data already stored using relational database technology. Therefore, information presented in the documents needs to be transformed into a specific ontology format, so, for example, the transformation of free text, web documents, and legacy database into the ontology level is the first step for the OMM. Tools are currently available for mapping the relational database (RDB) schemas on to resource description framework (RDF)/RDF schema (RDFS)/web ontology language (OWL), such as the Jena relational database [12] which is a declarative language to describe mappings between relational database schemata and OWL ontologies.

3.2.2 Ontology mapping rules

This step is to define and specify mapping rules between different ontologies. These mapping rules

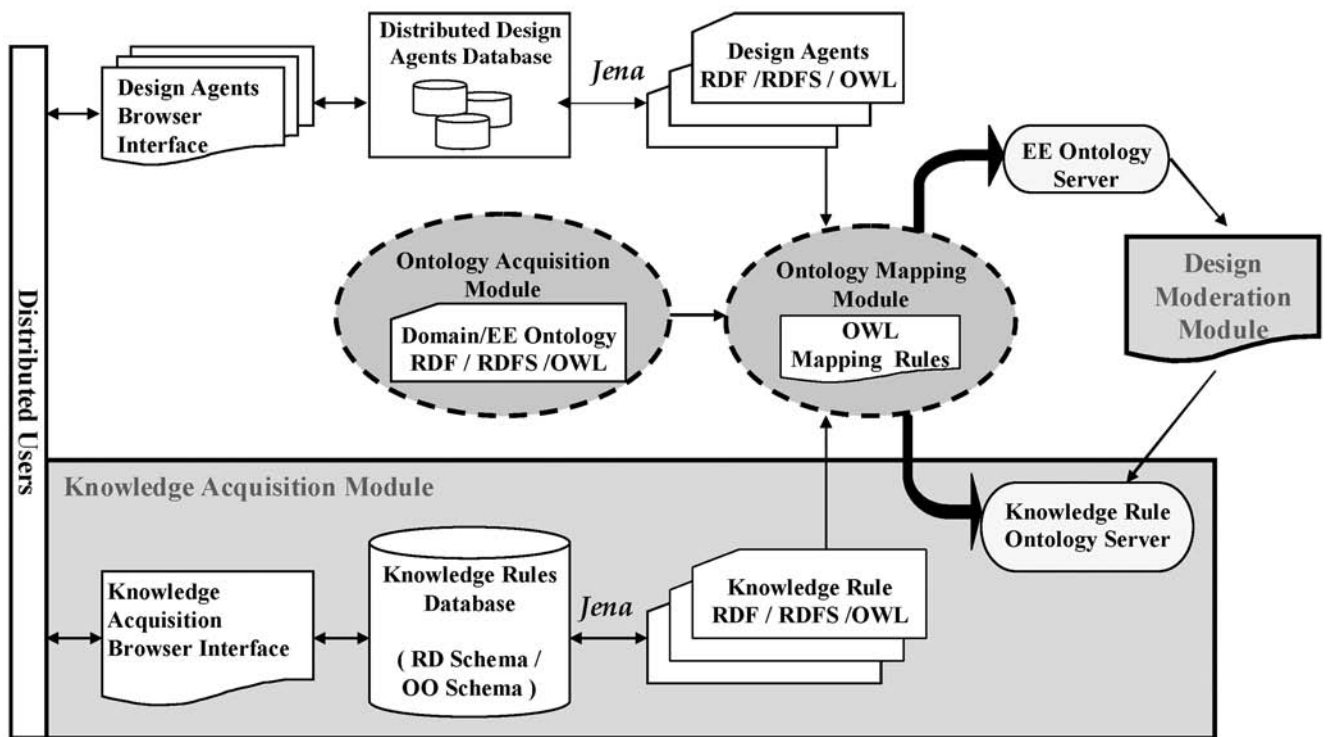


Fig. 5 The structure of the KAM and DMM in the EEMSEM

define how to transform source-ontology instances into target-ontology instances. Within the EEMSE moderator, all the individual MSE software's information must be normalized into the OWL primitives and be mapped to the mediating ontology, called the domain ontology, which is selected from the EE ontology of OAM. This is because OWL provides built-in ontology mapping support, for example, a particular class or property in one ontology is the same as a class or property in another ontology (`owl:sameAs`).

3.3 Knowledge acquisition module and design moderation module

In order to perform the moderation duties as mentioned in section 2, the moderator must retain and apply knowledge about the knowledge used by each project team member's MSE applications (these will be referred to as design agents here, to maintain consistency of terminology with the earlier MISSION MSE moderator research). Therefore, the moderator must be able to collect, or have access to, contact information and knowledge for each of the design agents. The KAM is used to create, delete or amend knowledge about what is important to any individual design agent. Therefore, it is important to modify this knowledge when new design agents join or if existing agents are changed significantly, resulting in changes to their associated knowledge which the EEMSEM uses to identify potential design conflicts. The knowledge structures repose in an

object-oriented knowledge rules database based on the knowledge representation model as in the MISSION MSE moderator.

However, the KAM in the EEMSEM would be translated into the neutral format (EE ontology) for dealing with any syntactic and semantic differences in the terminology that may be used by different project team members. This is achieved through the OAM and the OMM and then this knowledge about design agents can repose as mapped results in the knowledge rules ontology server, as shown in Fig. 5. Additionally, it is recommended that the KAM in the EEMSEM should be a web browser interface, so that the design agents could add, delete, or edit the knowledge rules about their interests 24 hours a day and seven days a week around the world.

The DMM is used to assist and keep track of changes made to the MSE design documents and identify whether any current design agent may be interested in the change. The change details should therefore also go through the translation process into the neutral format as described above for the KAM and the mapped result of the change details will be reposed in the EE ontology server, as shown in Fig. 5. Therefore the DMM should be activated whenever a change is made to any information that may be related to interests recorded in any design agent module. These changes can then be passed through the translation process, through the OAM and the OMM and into the EE ontology server. If information changes in the EE ontology server have

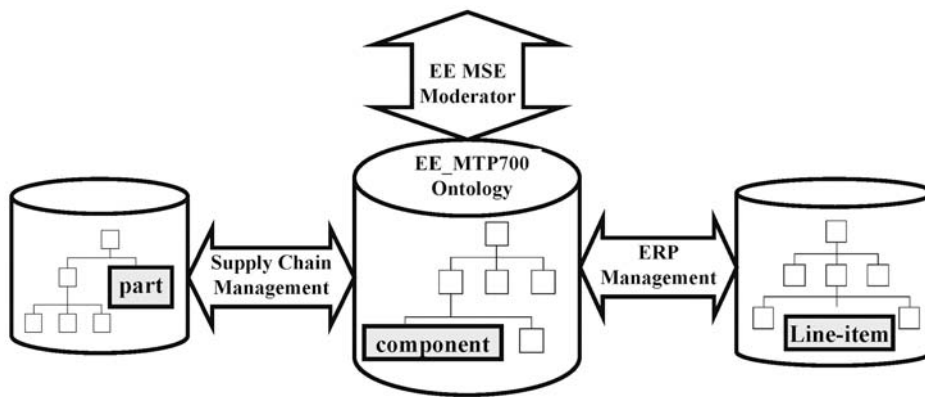


Fig. 6 Ontology mapping into the common ontology model

been identified, the DMM will be notified of change and also connected to the knowledge rule ontology server which is needed for the moderation process of conflict detection.

4 A CASE EXAMPLE

In order to illustrate the functionality of the EEMSEM, a manufacturing e-purchasing example study has been used. This is an extended team project, identified as the EE_MTP 700 project and includes factories in Motorola* Technology Malaysia and its collaborators in a two-way radio design and manufacture project. Case study examples were carried out by telephone interview, email correspondence, and web information from Motorola Technology Malaysia for this research, but in the following hypothetical examples, some values have been changed or assumed by the researchers to complete the scenarios. For ease of reference, Motorola Technology Malaysia will be referred to as Motorola in the following explanations.

This case example was used to demonstrate the conflict moderation work between the extended project teams' MSE agents (e.g. Motorola's enterprise resource planning (ERP) purchasing agent, Unitech's supply chain management (SCM) agents, and other MSE agents within this project). For example, Unitech's SCM is one of the participant systems in the extended project. As part of the extended project, Unitech's SCM determines that there should be a minimum quantities limitation of not less than 3000 units on their parts order. However, at some point during the operation of the extended project, there is a policy change in Motorola, for their ERP's purchase orders system that determines that the

electronic signature approval levels are reset to permit a maximum quantity on each line-item of 2000 units.

The EEMSEM here must be able to identify when the ERP's purchase agent changes the approval levels for the electronic signature in the quantity attribute of the line-item object as this change may cause conflict, hence the moderator must communicate the detection of this possible conflict to all interested MSE agents. When the above information change is made, the EEMSEM should identify that the Unitech SCM is the design agent that will be affected and that problems may occur with the quantity attribute of the part object. Therefore, the EEMSEM should issue an appropriate warning message to the Unitech SCM (e.g. via emails).

This EE project example shows that each company has its own processes, databases, information, and knowledge systems in place. Inevitably, each will also use its own languages and terminologies, which will have developed over a period of time through the company's working practices and experiences in particular industry sectors, the culture in the particular organization, and many other contributory factors. Each partner within the EE project will need to exchange and share some information and knowledge related to the project they are working on together, but this is inherently complex because they do not automatically work with a common language or common information models or structures. Figure 6 shows an example of two different identifiers existing with different models or databases, but having the same meaning. Hence, Motorola's ERP identifier, 'line-item' number and Unitech's SCM variable, 'part' number both have equivalent meanings in the purchasing process. They are therefore both mapped to 'component' in the agreed EEMTP700 ontology.

The role of the OAM and the OMM have therefore been proposed and experimental implementations undertaken to make the concept of knowledge and information integration possible, by providing an interoperability mechanism for dealing with the

* Motorola and the stylized 'M' logo are registered in the US Patent and Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Incorporated, 2004.

Table 1 OWL:sameAs axioms for semantically mapping into EE_MTP 700

Scm:part → ee_mtp700:component ← erp:line-item	Scm:quantity → ee_mtp700:quantity ← erp:quantity
<pre><owl:Class rdf:ID="scm.Part"> <owl:sameAs ref.resource="#ee_mtp700_Component"/> </owl:Class> <owl:Class rdf:ID="erp.Lineitem"> <owl:sameAs ref.resource="#ee_mtp700.Component"/> </owl:Class></pre>	<pre><owl:DatatypeProperty rdf:ID="scm.quantity"> <rdfs:domain rdf.resource="#scm_Part"> <owl:sameAs rdf.resource="#ee_mtp700:quantity"/> </owl:DatatypeProperty> <owl:DatatypeProperty rdf:ID="erp.quantity"> <rdfs:domain rdf.resource="#erp.Lineitem"> <owl:sameAs rdf.resource="#ee_mtp700:quantity"/> </owl:DatatypeProperty></pre>

above differences. It is assumed that the EE_MTP700 ontology derived from the proposed MSE ontology model has been chosen as the common/mediated ontology for the MTP 700 project. The ontology mapping to EE_MTP700 was undertaken using OWL primitive mappings as shown in Table 1.

The built-in OWL:sameAs statement links an individual to an individual that actually refer to the same thing: the individuals therefore have the same 'identity'. The OWL:sameAs axioms are often used in defining mappings between ontologies. In this case, the concepts from scm:part have the same meaning as the concepts from ee_mtp700:component. Moreover, the concepts from erp:line-item also have the same meaning as the concepts from ee-mtp700:component. The axioms should ensure that when someone queries the SCM for the instances of the

part, the result includes all instances of the component from the ee_mtp700. Also, the instances of the line-item will have the identity instances of the component from the ee_mtp700.

The EEMSEM must be able to acquire knowledge about individual design team members, the knowledge about what changes are important to them, and what actions should be taken if such changes occur. Figure 7 shows the interface from the KAM, which can be used whenever new agents join a project or existing agents are changed in any way.

In this case study example, Unitech have put their minimum part order quantities constraint into the KAM by creating a new design agent module called (SCM1). The SCM1 design agent module knows how to process its own knowledge, as this behaviour is implemented in methods of the various objects,

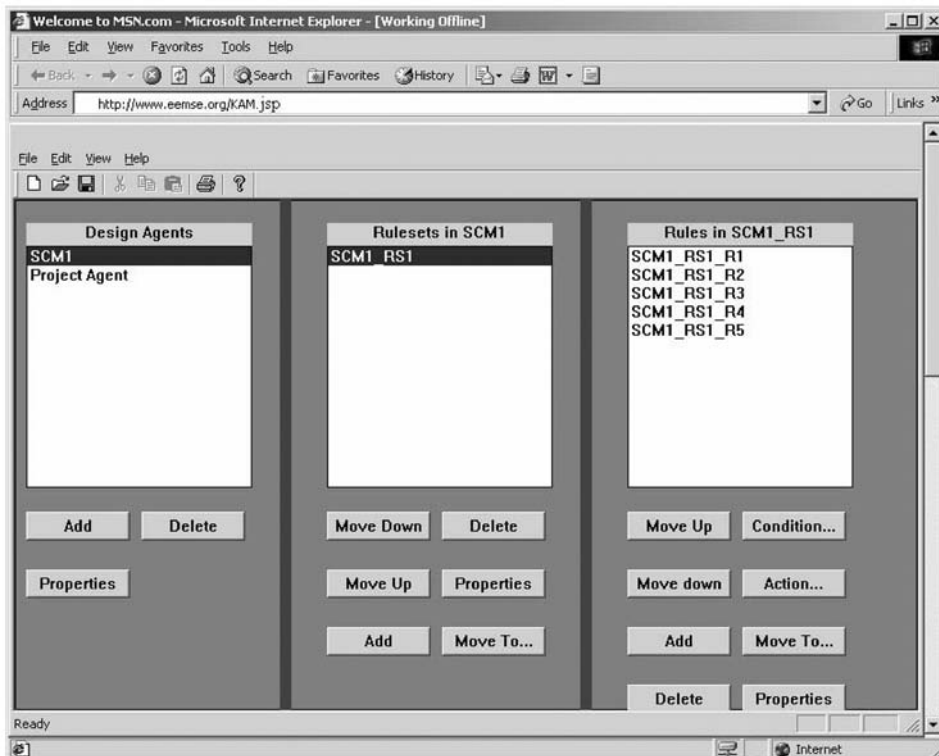
**Fig. 7** KAM for Unitech's SCM agent

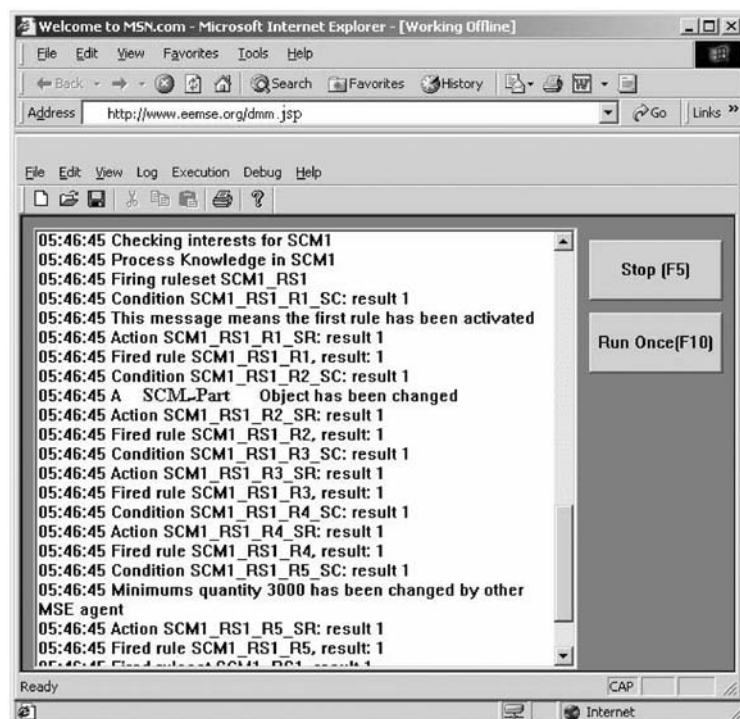
Table 2 The details of each rule in the SCM1_RS1 ruleset

Add a rule	Create condition and select type of condition	Detail of the condition	Create resulting action and select type of action	Detail of the action
SCM1_RS1_R1	Always true	–	Write message	This message means the first rule has been activated
SCM1_RS1_R2	Check change value	Type of value: class of changed object Value: SCM_Part	Write message	A SCM_Part object has been changed
SCM1_RS1_R3	Check change value	Type of value: attribute change Value: quantity	Put MSE value into memory	Attribute: quantity Type of value: int Working memory: in temp
SCM1_RS1_R4	Always true	–	Put literal into memory	Value: 3000 Type of value: int Working memory: in check
SCM1_RS1_R5	Working memory value equivalence	Type of equivalence: check > temp	Write message	Minimum quantity 3000 has been changed by other MSE agent

including 'ruleset', 'rule', 'condition', and 'action' objects (see Fig. 7). Each ruleset object can be associated with any number of rule objects. Each rule is associated with a condition object and a resulting action object. In the current case study, the SCM1_RS1 rulesets includes five rules, SCM1_RS1_R1, SCM1_RS1_R2, SCM1_RS1_R3, SCM1_RS1_R4, and SCM1_RS1_R5, which have been populated into the SCM1 design agent module, and which embody the SCM1's interests in details for the minimum part order quantities constraint. The details of each rule with its associated type of the condition and the type of the resulting action are

illustrated in Table 2 and saved into the object knowledge database.

So, assuming in this instance that information is changed at Motorola, resulting in the electronic signature approval levels being reset to 2000 units, the EEMSEM here must be able to identify when Motorola's ERP purchasing agent changes the approval levels for the electronic signature in the quantity attribute of the line-item object through his extranet browser interface, see Fig. 5. The EEMSEM will then pass and translate details of this change through the OAM and the OMM of the EEMSEM to recognize eventually that the information change is

**Fig. 8** The DMM in the EEMSEM

the quantity attribute of the component object in the EE_MTP700 ontology in the EE ontology server.

When the information change in the EE ontology server has been identified, the EEMSEM will then execute its DMM and also connect to the knowledge rule ontology server in which repose the mapped results from the OAM and the OMM of the knowledge rules that were acquired by the KAM earlier. The change information will then be processed to determine which, if any, of the participants in the extended project are interested in the change to the quantity attribute of the component object change. If the EEMSEM identifies any interested participants, it should then (still using the DMM) communicate the detection of the possible conflict to all the interested participants in the extended project (referred to here as MSE agents).

Figure 8 shows the operation of the DMM on this case study example; initially (on the top two lines) finding that Unitech's SCM1 agent is the one that will be interested. As the SCM1 stores the constraint on the quantity attribute of the part object (scm:part \equiv ee_mtp700:component) through OAM and OMM in the knowledge rule ontology server, which matches Motorola's ERP information change (erp:line-item \equiv ee_mtp700:component) in the EE ontology server. Therefore, the DMM identifies that SCM1 is the design agent affected when Motorola's ERP information change is detected in the EE ontology server, and then the DMM processes the knowledge in the SCM1 design agent module. Finally, a warning message 'minimum quantity 3000 has been changed by other MSE agent' should be sent to Unitech (e.g. via emails).

5 CONCLUSIONS

The EEMSEM framework is proposed as a possible way to integrate different information models via an ontology-mediated translation service. The MSE ontology model may involve simple logical reasoning for semantic and syntax mapping. However, the EEMSEM is limited to a knowledge-based approach for the extraction of useful information based on the established object oriented knowledge database. It does not involve the discovery of new knowledge, such as ontology learning, automatic knowledge

creation, and automatic knowledge retrieval by logical axioms. Future development is recommended, such as ontology-based information extraction towards automatic knowledge creation and knowledge retrieval by rules, logic, and proof, to improve and extend the application of the EEMSEM.

REFERENCES

- 1 **Berners-Lee, T.** *Semantic web road map*, 1998, available on-line as <http://www.w3.org/DesignIssues/Semantic.html>.
- 2 **Berners-Lee, T.** *Semantic web on XML*, 2000, in slides available on-line as <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>.
- 3 **Harding, J. A.** and **Popplewell, K.** Driving concurrency in a distributed concurrent engineering project team: a specification for an engineering moderator. *Int. J. Prod. Res.*, 1996, **34**(3), 841–861.
- 4 **Harding, J. A., Popplewell, K.,** and **Cook, D.** A manufacturing system engineering moderator: an aid for multi-discipline project teams. *Int. J. Prod. Res.*, 2003, **41**(9), 1973–1986.
- 5 **Lin, H. K.** and **Harding, J. A.** An ontology driven manufacturing system engineering moderator for global virtual enterprise teams. In *Advances in manufacturing technology XVII*, (Eds Y. Qin and N. Juster) 2003, pp. 365–370 (Professional Engineering Publishing Ltd, Bury St Edmunds and London, UK).
- 6 **Lin, H. K., Harding, J. A.,** and **Shahbaz, M.** Manufacturing system engineering ontology for semantic interoperability across extended project teams. *Int. J. Prod. Res.*, 2004, **42**(24), 5099–5118.
- 7 **Lin, H. K.** *Manufacturing system engineering ontology model for global extended projects team*. PhD Thesis, Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, UK, 2004.
- 8 **MISSION**, *Modelling and simulation environments for design, planning and operation of globally distributed enterprises*, ESPRIT IMS 29656 report, 1998–2001.
- 9 **Goranson, H. T.** *The agile virtual enterprise: cases, metrics, tools*, 1999, pp. 109–156 (Greenwood Press).
- 10 **Maedche, A., Motik, B., Stojanovic, L., Studer, R.,** and **Volz, R.** Ontologies for enterprise knowledge management. *IEEE Intelligent Systems*, 2003, **18**(2), 26–33.
- 11 **Noy, N. F.** and **Musen, M. A.** Ontology versioning in an ontology management framework. *IEEE Intelligent Systems*, 2004, **19**(4), 6–13.
- 12 **McBride, B.** Jena: A semantic web toolkit. *IEEE Internet Computing*, 2002, **6**(6), 55–59.