Semantic Web and Case-Based Reasoning for Engineering Change Case Retrieval

Hong Joo Lee and Sung Joo Park

Graduate School of Management, Korea Advanced Institute of Science & Technology
207-43, CheongryangriDong, DongdaemunGu, Seoul, 130-722
Tel: +82-2-958-3647, Fax: +82-2-958-3604, E-mail: {hjlee, sjpark}@kgsm.kaist.ac.kr

Abstract

Development process for complex products such as automobile involves engineering changes that require redesign or adjustment of the products. Efficient management of engineering change is crucial for the productivity of new product development. Despite the importance of reusing past engineering change case, extant supporting systems remain mostly as document management or workflow systems that are insufficient in the reuse of various knowledge in the engineering change. The purpose of this paper is to design a case retrieval mechanism to facilitate the reuse of the knowledge generated in the engineering change. Past engineering change cases are represented using Semantic Web technology and Case-Based Reasoning (CBR) is used to retrieve relevant cases with domain ontologies. Similarities of retrieved cases and target cases are simulated with other various mechanisms.

Keywords:
Case-Based Reasoning, Semantic Web, Engineering Change

1. INTRODUCTION

Engineering Changes (ECs) are the changes or modifications of products and constituent components after the product design is released [9, 18, 39]. Development processes for complex engineering products involve many ECs to reflect technological developments, resolve conflicts in the design, and improve the overall quality of the products [1, 31, 39]. ECs are inevitable in complex products because the development usually takes a long time and accompanies collaboration among designers and engineers in a distributed environment [18].

Efficient management of engineering change is crucial for the productivity of new product development. Despite the importance of reusing past engineering change case, extant supporting systems remain mostly as document management or workflow systems that are insufficient in the reuse of various knowledge in the engineering change.

The purpose of this paper is to design a case retrieval mechanism to facilitate the reuse of the knowledge generated in the engineering change. The collaboration model which is a meta model of knowledge item and ontologies are used to represent EC cases. We define ontologies for the five dimensions of ECM(Engineering Change Management): product, component, problem, process, and solution.

The suggested approach is applied to ECs of automobile development in a major Korean company. We performed a case study to derive specific requirements for the problem domain in collaboration with the company. The collaboration model and the CBR mechanism are explained with illustrative examples from the company.

The remainder of this paper is organized as follows. Section 2 presents a literature review on ECM and knowledge management (KM). Section 3 describes the collaboration model and examples of knowledge representation and retrieval. Section 4 presents experiments of case retrieval and their performances. Section 5 presents a discussion and conclusion.

2. BACKGROUND OF THE RESEARCH

2.1 Engineering Change Management

ECs are not always to the detriment of the development project, as many cost savings or performance improvements are brought into the project in the form of ECs [9, 10, 39]. Table 1 shows a summary of the major causes of ECs from related literature. We can see from Table 1 that, although there are unnecessary changes that should be avoided, many of the ECs are actually helpful and, thus, it is neither desirable nor unrealistic to focus our efforts on just eliminating the ECs [9]. Consequently, what is more important is the efficient management of ECs and EC processes to reduce time and costs.

Table 1 – Causes of Engineering Changes

Causes of Engineering Changes	Descriptions
Careless Mistakes	Corrections of errors on a document
Poor Communication	Faults in the interpretation of customer demands into technical requirements Changes in the customer specification Changes in the manufacturing process or situations
Snowballing Change	Change of a part depending on altered function or production requirements

	Organizational, technological and operational changes
Cost Savings	Cost savings Change, replacement, withdrawal, and introduction of a part
Ease of Manufacturing	Difficulties in parts fabrication or assembly
Product Performance Improvement	Weakness in the product identified during prototype testing Quality problems with some subsystems or component Development for future product revisions

The formal process of EC is usually composed of four stages [17, 18, 31, 39]: raising an Engineering Change Request (ECR), evaluating requested changes, issuing ECOs to relevant employees, and storing and analyzing ECOs for management purposes (Figure 1).

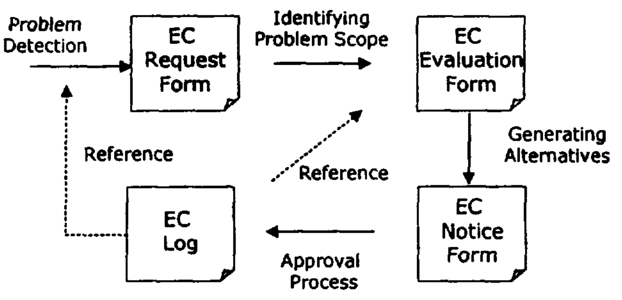

Figure 1 - The Process of Engineering Change Management

2.2 Knowledge management in ECM

Many studies on knowledge management in ECM and product development can be classified into two categories. The first, behavioral research, investigates KM practices and their relation to the performance of ECM and product development [15, 19, 23, 24, 37]. These studies show that product developments are knowledge intensive activities, and learning of past experiences is critical to the performance of the ECM and product development [19, 23, 24, 37]. Madhavan and Grover (1998) argued that the creation of new knowledge can be viewed as the central theme of the NPD process. The NPD manager's task is to manage the transition from NPD team's embedded knowledge to new product's embodied knowledge [24]. Lynn et al. (2000) found several factors related to KM for reducing cycle time of product developments and improving their probabilities of success: documentation of project information, storage and retrieval systems for project information, information reviewing practices. These factors are related to whole knowledge processes including knowledge creation, organization & coordination, and reuse. Also, KM is emphasized for handling ECs. Saeed et al. (1993) suggested that organizations should manage the development of focused manufacturing knowledge to avoid many manufacturing-related ECs.

The second, technical research provides KM supporting systems for product development [29, 32, 40, 42]. These studies suggest a number of KM supporting systems for product development to capture collaborative process knowledge and share product data [25, 26, 29, 30, 32, 42]. Ramesh and Tiwana (1999) focused on providing support for a collaborative task with emphasis on capturing process knowledge in collaborative systems. This research proposed to represent knowledge with concepts in product developments and concept map which describes dependencies of concepts [32]. This model is similar to the issue based information system (IBIS) model of argumentation [11] that has been used successfully in a wide range of domains to represent complex problem solving processes. Many studies introduced computer supported co-operative work (CSCW) approach for product development and evaluated their effectiveness. The productivity of engineering design teams can be further enhanced when the CSCW functions are augmented with integrated process development architecture [29]. TEAM Demonstrator is a virtual team working environment for the European automotive industry [25, 26]. It provided multimedia conferencing, application sharing, product library, and product data visualization tools. All companies involved in TEAM Demonstrator considered main benefits of using it as: identifying and resolving problems early, and time reduction of product development. Communication among engineers participating in product developments is also emphasized in the research. Numata and Taura (1996) emphasized a knowledge network in order to enhance the convenient transfer of knowledge among engineers.

3. Case Representation of Engineering Change

3.1 Design of the Collaboration Model for Engineering Change Management

The collaboration model was derived by considering the characteristics of ECs discussed in section 2. Another important perspective that was considered in the collaboration model is the perspective from the virtual collaboration, especially in regard to knowledge management. According to Ahn et al. (2004), the virtual collaborative work perspective should consider the organizational, activity, and context perspectives for bridging virtual collaborative work and knowledge management. Requirements of the organizational perspective are that the collaboration model should support the lifecycle of virtual collaboration and also ad hoc collaboration. The activity perspective requires natural alignment of knowledge creation and collaborative activities, and the context perspective emphasizes the preservation of contextual knowledge that consists of *organization*, *person* and *activity* which are critical in virtual collaboration environment [3].

The collaboration model is depicted in Figure 2 and the main features of the collaboration model are explained in the following:

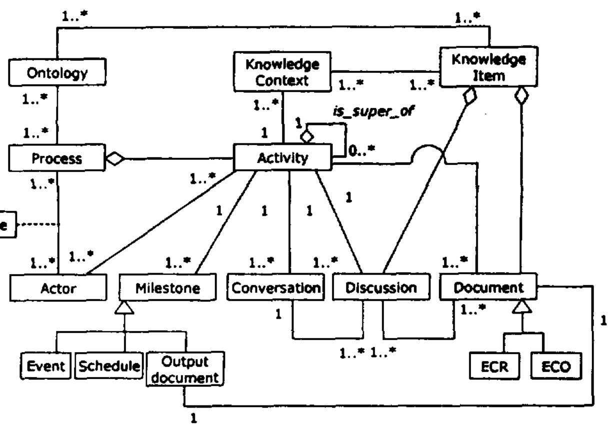

Figure 2 – The Collaboration Model

Activity and Process

A process can have hierarchically organized activities and the lifecycle status of an EC process as its attributes. After the completion of an EC, the whole set of knowledge items and the structure of activities in the process can be stored in the knowledge repository along with the knowledge context. The activity is a crucial entity of the collaboration model for integrating knowledge items with knowledge context. Knowledge items are associated with activities and other context entities are also associated with the activity entity.

Conversation

Conversation is a set of structured messages based on speech-act theory for supporting cooperative processes [2, 13]. An actor can initiate conversations in an activity and a commitment is negotiated through a procedure between a doer and a referent. During a conversation process, knowledge items are created and the status of an associated activity or document is updated.

Two types of Knowledge items: Document and Discussion

The collaboration model supports two types of knowledge with the document and discussion entities. The document entity represents formal and structured output documents such as ECR, ECO, and interim or final reports of an EC activity. The discussion entity represents unstructured communications between actors, and is associated with activity, document, and conversation entities.

Knowledge Context

The knowledge context is background and contextual information for knowledge items in ECs that are used to annotate them and to provide rich navigation paths. The knowledge context is mainly associated with the activity because the activity entity is at the very center of the collaboration model playing the role of integrating all of the constructs in the collaboration model [3].

Ontology

The ontology provides shared terminology of concepts and their relationships for various activities in ECs. For the automobile development project used as an example of this paper the following domain ontologies were considered (adapted and revised from [16]):

- Product ontology: a hierarchy of product segmentations and their instances such as passenger cars, recreation vehicles and commercial vehicles.
- Component ontology: a hierarchy of components, modules, and functions in a vehicle such as engine, cockpit, and axle.
- Problem ontology: a collection of problem types that describes causes of problems and reasons for ECs such as product safety and manufacturing difficulties.
- Solution ontology: a hierarchy of solution types in ECs such as product form modification and assembly hole relocation.
- Process ontology: the structure of a new product development process and its hierarchy of activities in a project such as planning, styling, and pilot production stages.

3.2 Knowledge Representation with the Collaboration Model and Ontologies

The collaboration model can be both a meta model of knowledge items in ECs and their context information at the same time. All of the constructs in the collaboration model are used to annotate the knowledge items (e.g., documents, discussions) along with their context information. Figure 3 shows how the collaboration model and ontologies are used to represent knowledge items. RDF Schema (RDFS) is a schema specification language developed for representing the RDF statements [38, 39] used widely for the Semantic Web [5, 12, 33]. An RDF annotation consists of a set of statements, each one specifying the property of a resource [12, 14, 33]. We generated RDFS for the collaboration model and the ontologies (see Appendix A and B for examples). The RDFS definition for the collaboration model was generated from the UML-based description of the model by using an automatic transformation tool Xpetal [27, 28, 41]. Based on the RDFS descriptions, instances of domain ontologies and the entities of the collaboration model are generated. The RDF instances of the domain ontologies associated with the definition of the collaboration model are also used as the definition of cases used by the CBR mechanism. Thus, the knowledge items in EC processes are annotated by the RDF definition for such outputs as ECOs, modification results, and problem solving reports.

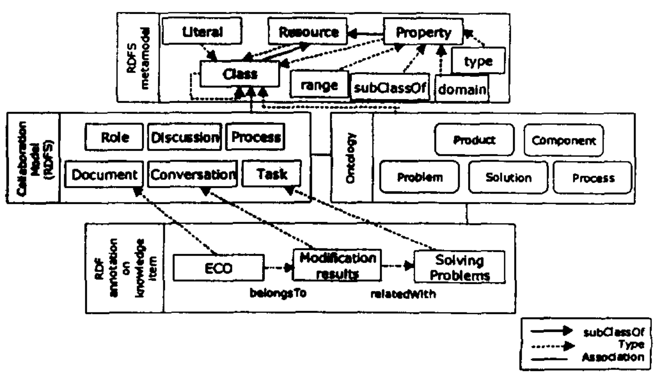

Figure 3 – Knowledge Representation

Other systems such as a product data management (PDM) system and problem management system, can share knowledge items from ECM because what a knowledge item means can be reasoned through knowledge annotation of the item. Also, knowledge annotation provides further navigation paths through the link with their contextual information.

3.3 Knowledge Reuse in the Engineering Change Management

Many ontology driven knowledge management initiatives use query languages to retrieve knowledge items [6, 20]. Ontology driven KM applications answer queries by finding a relevant set of facts that are the logical consequences of the ontology and the set of instances, but the strictness of deductive reasoning approaches has been recognized as one of the major obstacles in weakly structured problems [4].

CBR can relax the strictness by allowing the retrieval of knowledge items that are close to queries, if not exact matches [4]. CBR systems process a new problem at hand by retrieving a most similar case from memory, adapting the case by adjusting and modifying discrepant attributes, and storing the new case in the memory [22]. However, it is difficult to calculate the similarity between complex products and problems because they often involve a number of parts or components, and the attributes of the parts or components cannot be easily weighted numerically. For example, consider the similarity between a 2. 5 DOHC gasoline engine and a 2.0 diesel engine. In this example, domain experts will usually assign the weight values and construct a similarity measure utilizing his/her expertise. However, the weights and similarity measures can be easily outdated for new products or new components, and also, there are numerous types of different cases that make the manual effort of experts often infeasible. Therefore, relieving the maintenance efforts for similarity measure by automation is a critical issue in CBR when the cases are complex and involve various types of different constructs [21].

According to Resnik (1999), informativeness of a concept in an IS-A taxonomy can be used to measure concept similarities. Let the taxonomy be augmented with a function $p: C \rightarrow [0,1]$, such that for any $c \in C$, $p(c)$ is the probability of encountering an instance of concept c. This implies that p is monotonically non-decreasing as one moves up the taxonomy. The information content of a concept c can be quantified as negative the log likelihood, $-\log p(c)$ [8, 35]. As probability of encountering an instance of concept c increases, informativeness decreases; so the more abstract a concept, the lower its information content. The information shared by two concepts is indicated by the information content of the concepts that subsume them in the taxonomy. Formally, we can define

$$sim(c_1, c_2) = \max_{c \in S(c_1, c_2)} [-\log p(c)], \qquad (1)$$

where $S(c_1, c_2)$ is the set of concepts that subsume both c_1 and c_2. For an example, let us consider how the similarity between 'screwing' and 'fitting' in the problem ontology in Figure 4 can be computed. 'Assembly' and 'Problem' concepts are members of $S(screwing, fitting)$. Because the information content of a concept 'problem' is 0, the similarity between 'screwing' and 'fitting' is the information content of a concept 'assembly' (1.2039).

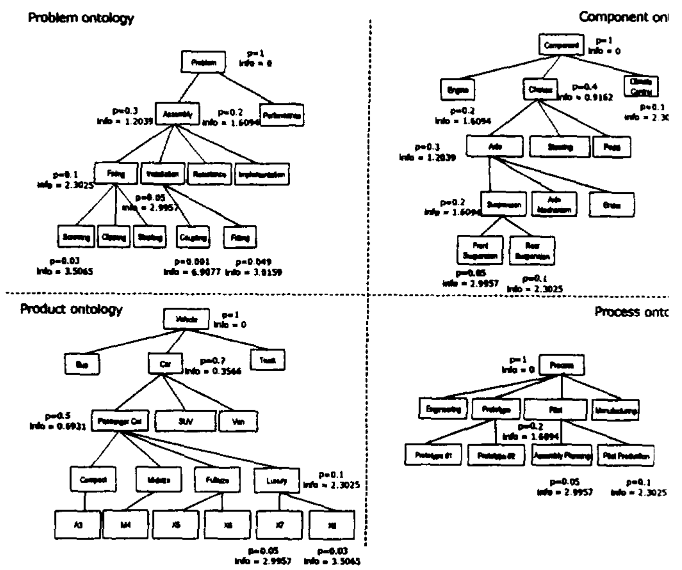

Figure 4 – Fragment of Ontologies

One of main issues in similarity measure based on information content is to calculate the probability of encountering an instance of concept, $p(c)$. Frequency of problems in a problem type in past ECs can be a concept probability in the problem ontology. Formally, we can define

$$freq(c) = \sum_{n \in Problem(c)} count(n), \qquad (2)$$

where Problem(c) is the set of problems subsumed by concept c. Concept probabilities can be computed simply as a relative frequency:

$$p(c) = \frac{freq(c)}{N}, \qquad (3)$$

where N is the total number of problems observed. This formulation assigns probability 1 to the top concept of the taxonomy, leading to the desirable consequences that its information content is zero [8, 35]. Also, frequency of products in a product type can be used for product ontology. The number of sub components in a component type, number of ECs in development stages, and the number of solutions in a solution type can be used for component ontology, process ontology and solution ontology respectively.

Now, because the cases should be associated with multiple ontologies, similarities between cases can be calculated by a weighted sum of concepts similarities in ontologies. Formally, we have

$$Sim(C_1, C_2) = \frac{\sum_i w_i \times f_i \times sim_i(c_{1i}, c_{2i})}{\sum_i w_i}, \quad (4)$$

where C_1, C_2 are cases, w_i is the weight of ontology i, sim_i is a similarity between concepts of cases, C_1, C_2, in ontology i. The weight of ontologies can be given by domain experts and also can be decided through the Analytic Hierarchy Process comparing key criteria pair-wisely [7, 36]. f_i is the inverse concept frequency of ontology i. Ontology structure, such as number of concepts and depth level, has an influence on concept similarities. Complex ontology with many concepts usually has higher similarities than simple ontology. To reduce this influence on concept similarities, we introduce inverse concept frequency as similar as inverse document frequency in document indexing [38] and f_i can be calculated as following:

$$f_i = \log \frac{M}{N_i}, \quad (5)$$

where M is the total number of concept through all ontologies and N_i is the number of concept in ontology i.

This approach based on information content has a number of advantages in managing similarity measure. First, concept probability of ontologies can be configured from case bases simply. Second, concept probability can be maintained and updated easily when new cases are added or past cases are deleted. If we are able to add domain ontologies of document and activity type, concept similarity based CBR approach can be expanded to retrieve relevant documents or activities.

4. EXPERIMENT OF CASE RETRIEVAL

We collected 261 past EC cases of a development project from a Korean automobile. An experiment of case retrieval was performed by selecting 5 similar cases from other cases for one case respectively.

Since the retrieval results from the CBR mechanism will be heavily dependent upon the weights, the Analytic Hierarchy Process (AHP) was utilized. A set of pair-wise comparison data of the ontologies was collected from the experts in a Korean automobile company using questionnaires. The result of AHP calculation is presented in Figure 5.

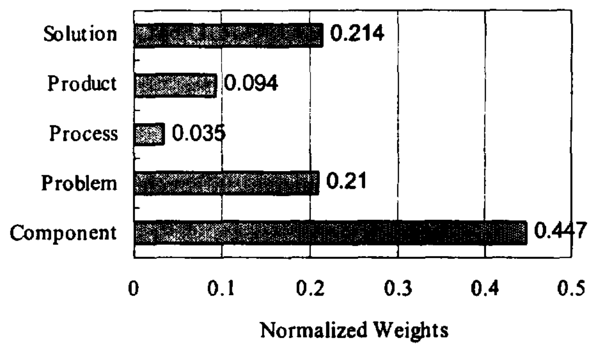

Figure 5 - Weights of the five ontologies

We compared following three methods for EC case retrieval: component based retrieval, ontology based retrieval, and keyword based retrieval.

Component based retrieval: This is a simple mechanism used by the Korean automobile company. This mechanism finds recent cases happened on same engineering components.

Ontology based retrieval: This is an ontology driven CBR mechanism with concept based measure (suggested in Section 3). We established small domain ontologies for the experiment from collected EC cases. Normalized weights on ontologies are used to calculate similarities between cases using ontologies and concept based measure.

Keyword based retrieval: We extracted keywords from all cases and eliminated keywords under predefined TF-IDF threshold level. We made keyword vectors using for each case. Then, we calculated similarities between cases using cosine measure.

Similarities between a target case and retrieved cases are calculated using Formula (4) with the weights in Figure 5. Average similarities of experiment methods are presented in Table 2.

Table 2. Performance of Case Retrieval

	Total Number of similar cases	Same product	Same process	Same problem	Same component	Average Similarity
Component	1204	1198	668	211	1204	0.2281
Ontology	1305	1289	1003	697	934	0.271
Keyword	1305	1268	838	616	202	0.1014

Component based retrieval can't provide 5 similar cases if number of same component cases is under 5. So, the total number of similar cases of component based retrieval is not 1305 (261×5). Next columns are the number of cases which belong to same product, process, problem, and component. We can see that ontology based retrieval mechanism provides more similar cases even if they are not exactly match with target cases' component.

5. DISCUSSION AND CONCLUSION

5.1 Discussion

The proposed model offers a number of advantages through the engineering change lifecycle. First, the Semantic Web provides a means to represent and share various types of engineering change knowledge in geographically distributed environments, and also to link the knowledge with contexts in various stages of engineering change activities. Explicit definitions of knowledge using ontologies are also helpful for accumulating engineering change cases for reuse. Second, relevant cases are stored and retrieved by using the CBR technique for providing insights from past engineering changes. The concept-based similarity measure used in the CBR mechanism enables efficient maintenance of the retrieval process without excessive manual efforts of domain experts. The retrieved cases, along with the knowledge context, provide effective navigation paths for the set of relevant knowledge items.

5.2 Conclusion

There are three significant points in this paper: First, we showed why accumulation and reuse of knowledge are crucial in ECM by reviewing related research and analyzing the case study. Second, the collaboration model for ECM was designed to tie collaborative activities to knowledge management through EC lifecycles. Third, we presented a means to represent and reuse EC knowledge by the CBR mechanism where the cases are defined with the collaboration model and domain ontologies.
However, limitations and further research issues remain. We used the concept-based similarity measure for calculating similarities, since concepts in EC cases can be hierarchically organized. Although this measure provides a strong advantage where attributes in complex cases cannot be weighted precisely, further refinement of the measure reflecting the characteristics of products and components is needed. In addition, empirical evaluation of CECM and the collaboration model for practical usefulness is considered as an interesting future research topic.

REFERENCES

[1] P. S. Adler, K. B. Clark, "Behind the Learning Curve: A Sketch of the Learning Process," *Management Science*, vol. 37, no. 3, pp. 267-281, 1991.
[2] A. Agostini, G. De Michelis, M. A. Grasso, "Rethinking CSCW systems: the architecture of Milano," *Proceedings of the Fifth European Conference on Compüter Supported Cooperative Work*, pp. 33-48, 1997.
[3] H. J. Ahn, H. J. Lee, K. H. Cho, S. J. Park, "Utilizing Knowledge Context in Virtual Collaborative Work," *Decision Support Systems*, Special issue on Collaborative Work and Knowledge Management in Electronic Business, Forthcoming.
[4] R. Bergmann, M. Schaaf, "Structural Case-Based Reasoning and Ontology-Based Knowledge Management: A Perfect Match?," *Journal of Universal Computer Science*, vol. 9, no. 7, pp. 608-626, 2003.
[5] T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web," *Scientific American*, May 2001.
[6] J. Broekstra, A. Kampman, F. Harmelen, "Sesame: A generic Architecture for Storing and Querying RDF and RDF Schema," in J. Davies, D. Fensel, F. Harmelen (Eds.), *Towards the Semantc Web: Ontology-Driven Knowledge Management*, West Sussex: John Wiley & Sons, pp. 71-89, 2003.
[7] C. Chen, C. Huang, "A multiple criteria evaluation of high-tech industries for the science-based industrial park in Taiwan," *Information & Management*, vol. 41, pp. 839-851, 2004.
[8] J. Choi, M. Cho, S. Park, P. Kim, "Concept-Based Image Retrieval Using the New Semantic Similarity Measurement," in V. Kumar et al. (Eds.), ICCSA 2003, LNCS 2667, pp. 79-88, 2003.
[9] K. B. Clark, T. Fujimoto, *Product Development Performance: Strategy, Organization, and Management in the World Auto Industry*, Boston: Harvard Business School Press, 1991.
[10] K. B. Clark, S. C. Wheelright, *Managing New Product and Process Development: Text and Cases*, New York: Free Press, 1993.
[11] J. Conklin, M. Begeman, "gIBIS: a hypertext tool for exploratory policy discussion," *ACM Transactions on Office Information Systems*, vol. 6, no. 4, pp.303-331, 1988.
[12] S. Decker, P. Mitra, S. Melnik, "Framework for the Semantic Web – An RDF Tutorial," *IEEE Internet Computing*, vol. 4, no. 6, pp. 68-73, 2000.
[13] G. De Michelis, M. A. Grasso, "Situating conversation within the language/action perspective: the Milan Conversation Model," *Proceedings of the 5th conference on CSCW*, pp. 89-100, 1994.
[14] A. Delteil, C. Faron-Zucker, R. Dieng, "Learning ontologies from RDF annotations," *Proceedings of the Second Workshop on Ontology Learning*, 2001.
[15] B. Fagerström, L. Olsson, "Knoweldge Management in Collaborative Product Development," *Systems Engineering*, vol. 5, no. 4, pp. 274-285, 2002.
[16] J. Golebiowska, R. Dieng-Kuntz, O. Corby, D. Mousseau, "Building and Exploiting Ontologies for an Automobile Project Memory," *Proceedings of K-CAP'01*, 2001.
[17] G. Q. Huang, W. Y. Yee, K. L. Mak, "Development of a web-based system for engineering change management," *Robotics and Computer Integrated Manufacturing*, vol.17, pp. 255-267, 2001.
[18] G. Q. Huang, W. Y. Yee, K. L. Mak, "Current practice of engineering change management in Hong Kong manufacturing industries," *Journal of Materials Processing Technology*, vol.139, pp. 481-487, 2003.
[19] A. C. Inkpen, "Creation Knowledge through Collaboration," *California Management Review*, vol. 39, no. 1, pp. 123-140, 1996.
[20] G. Karvounarakis, A. Magkanaraki, S. Alexaki, V.

Christophides, D. Plexousakis, M. Scholl, K. Tolle, "Querying the Semantic Web with RQL," *Computer Networks and ISDN Systems Journal*, vol. 42, no. 5, pp. 617-640, 2003.
[21] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann Publishers, 1993.
[22] M. M. G. Lorenzo, R. E. B. Perez, "A model and its different application to case-based reasoning," *Knowledge-Based Systems*, vol. 9, no. 7, pp. 465-473, 1997.
[23] G. S. Lynn, R. R. Reily, A. E. Akgün, "Knowledge Management in New Product Teams: Practices and Outcomes," *IEEE Transactions on Engineering Management*, vol. 47, no. 2, pp. 221-231, 2000.
[24] R. Madhavan, R. Grover, "From Embedded Knowledge to Embodied Knowledge: New Product Development as Knowledge Management," *Journal of Marketing*, vol. 62, no. 4, pp. 1-12, 1998.
[25] A. May, C. Carter, S. Joyner, "Virtual Team Working in the European Automotive Industry: User Requirements and a Case Study Approach," *Human Factors and Ergonomics in Manufacturing*, vol. 10, no. 3, pp. 273-289, 2000.
[26] A. May, C. Carter, "A case study of virtual team working in the European automotive industry," *International Journal of Industrial Ergonomics*, vol. 27, pp. 171-186, 2001.
[27] S. Melnik, "Representing UML in RDF," http://www-db.stanford.edu/~melnik/rdf/uml/, 2000.
[28] S. Melink, S. Decker, "A Layered Approach to Information Modeling and Interoperability on the Web," *Proceedings of ECDL'00 Workshop on the Semantic Web*, Lisbon, Portugal, 2000.
[29] L. Monplaisir, "An integrated CSCW architecture for integrated product/process design and development," *Robotics and Computer-Integrated Manufacturing*, Vol. 15, pp. 145-153, 1999.
[30] J. Numata, T. Taura, "A Case Study: A Network System for Knowledge Amplification in the Product Development Process," *IEEE Transactions on Engineering Management*, vol. 43, no. 4, pp. 356-367, 1996.
[31] P. Pikosz, J. Malmqvist, "A Comparative Study of Engineering Change Management in Three Swedish Engineering Companies," *Proceedings of DETC'98*, 1998.
[32] B. Ramesh, A. Tiwana, "Supporting Collaborative Process Knowledge Management in New Product Development Teams," *Decision Support Systems*, vol. 27, pp. 213-235, 1999.
[33] RDF, http://www.w3.org/TR/rec-edf-syntax/, 1999.
[34] RDFS, http://www.w3.org/TR/rdf-schema/, 2000.
[35] P. Resnik, "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language," *Journal of Artificial Intelligence Research*, vol.11, pp. 95-130, 1999.
[36] T. L. Saaty, *The Analytic Hierarchy Process*, New York: McGraw-Hill, 1980.
[37] B. I. Saeed, D. M. Bowen, V. S. Sohoni, "Avoiding Engineering Changes Through Focused Manufacturing Knowledge," *IEEE Transactions on Engineering Management*, vol. 40, no. 1, pp. 54 – 59, 1993.
[38] G. Salton, M. McGill, *Introduction to Modern Information Retrieval*, New York:McGraw-Hill, 1983.
[39] C. Terwiesch, C. H. Loch, "Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Development," *Journal of Product Innovation Management*, vol.16, pp.160-172, 1999.
[40] A. Tiwana, B. Ramesh, "A design knowledge management system to support collaborative information product evolution," *Decision Support Systems*, vol. 31, pp. 241-262, 2001.
[41] Xpetal, http://www.langdale.com.au/styler/xpetal/.
[42] S. B. Yoo and Y. Kim, "Web-based knowledge management for sharing product data in virtual enterprises," *International Journal of Production Economics*, vol. 75, pp. 173-183, 2002.