Ontology-Based Knowledge Integration for Distributed Product Knowledge Service*

Yuh-Min Chen[1], Yuh-Jen Chen[2], Chiung-Cheng Wen[3], and Hui-Chuan Chu[4]

***Abstract* — Conventional business models focus on providing end products and services regardless of product- or service-related knowledge. Such models are unable to meet customer requirements in the age of knowledge economy. This paper presents an ontology-based knowledge integration approach that is able to handle heterogeneity and variation of knowledge form dynamic distributed knowledge sources so as to support distributed product knowledge service in virtual product and process development.**

A distributed product knowledge service model was first proposed, which was then followed by design of a web service based framework for knowledge integration. An ontology-based meta-knowledge schema is proposed for defining local knowledge repositories. The technology of semantic similarity comparison is developed to solve the problem of semantic heterogeneity and information variation in ontology integration. The proposed knowledge integration approach is general and is directed towards realizing the concepts of product service, and thus achieving the goals of virtual product and process development.

***Index Terms* — Knowledge management, knowledge integration, virtual enterprise, ontology.**

I. Introduction

Virtual product and process development (VPPD) is viewed as one of the most promising business strategies, which can address global competition [1]. The essence of virtual product and process development is the integration of business activities and resources from different business units to satisfy quickly and efficiently customer needs from within a growing worldwide dynamic and competitive markets.

In the age of knowledge economies, besides products and services, enterprises are required to provide intangible product knowledge to meet the increasing requirements of product knowledge use in product lifecycle [2][3][4]. However, due to the variety and heterogeneity of product knowledge, effective management, integration and sharing of product-related knowledge throughout activities within a product lifecycle and its supply chain is a challenge in product and process development.

Product data and knowledge are involved in activities of product and process development; therefore, models and methods were developed for product data management [5][6][7][8]. However, most of them focused primarily on data and document management and seldom dealt with management of product knowledge.

* This research is financially supported by National Science Council of the Republic of China under Contract No: 96-2221-E-006-276-MY3.

[1]Yuh-Min Chen is with the National Cheng Kung University, Tainan, Taiwan, ROC (phone: 886-6-2757575#34222; fax: 886-6-2085334; e-mail: ymchen@mail.ncku.edu.tw).

[2]Yuh-Jen Chen is with the National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, ROC (e-mail: yjchen@ccms.nkfust.edu.tw).

[3]Chiung-Cheng Wen is with the National Cheng Kung University, Tainan, Taiwan, ROC (e-mail: wildcat@cacerl.ime.ncku.edu.tw).

[4]Hui-Chuan Chu is with the National University of Tainan, Tainan, Taiwan, ROC (e-mail: huichu@mail.nutn.edu.tw).

Research efforts have also been made on development of information sharing systems and collaborative tools to support inter-organizational processes and teamwork [1][9][10]. However, only a few works addressed supporting knowledge integration in virtual product and process development context. Knowledge integration in virtual product and process development has been a technical challenge due to the fact that it is highly dependent on its model and the business knowledge is rather heterogeneous with semantic variations [11][12].

This paper presents an ontology-based knowledge integration approach that is able to handle heterogeneity and variation of knowledge form dynamic distributed sources. The characterization and modeling of product knowledge service are first conducted, which is followed by design of a web service based framework for knowledge integration. An ontology-based meta-knowledge schema is proposed for defining local knowledge repositories. The technology of semantic similarity comparison is developed to solve the problem of semantic heterogeneity and information variation in ontology integration. The proposed knowledge integration approach is general and is directed towards realizing the concepts of product knowledge service, and thus achieving the goals of virtual product and process development.

II. Product Knowledge Service Modeling

The objective of characterization and modeling is to formally describe the behavior of product knowledge service to pave the way for development of knowledge integration framework.

2.1 Concept, Definition and Characterization

It is believed that effective use of product-related knowledge, such as knowledge and experience related to product design and manufacturing, and knowledge of optimum product usage, can significantly increase efficiency of product and process development. To achieve the goal, product-related knowledge that develops gradually through product lifecycle should be accumulated and provided at right time through a product knowledge service sharing mechanism. Accordingly, product knowledge service can be defined as: "to acquire, store, integrate and apply related product knowledge throughout activities in product lifecycle to ultimately satisfy customer requirements."

The product knowledge service model has the following four characteristics: (1) distributed: enterprises involved in product lifecycle may locate in geographically disparity and, therefore, form a distributed product knowledge service environment, (2) hierarchical: activities in a product lifecycle can be further decomposed into sub-activities according to specialty, which causes a product knowledge service model having a hierarchical structure, (3) flexible: allied enterprises may dynamically join and withdraw from a product knowledge service team based on their own concerns, and (4) product-oriented: product knowledge service is associated with a product lifecycle and its supply chain.

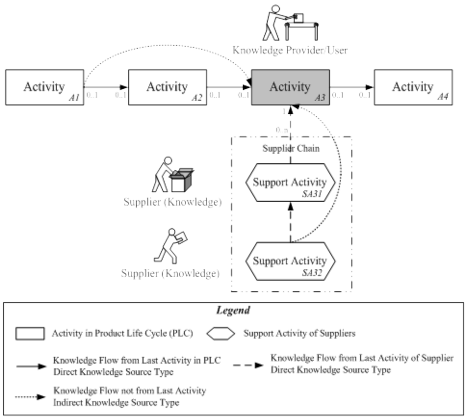

Fig. 1. Generic model for product knowledge service.

2.2 Product Knowledge Service Modeling

Fig. 1 shows a generic model for product knowledge service. The horizontal axis represents product lifecycle activities, and the vertical axis is the activity supply chain. In this generic model, each product lifecycle activity has the following characteristics:

(1) Zero, one or many pre- and post-activities comprise the activity flow in a product lifecycle.

(2) One or many input items and output items form the item flow for a product lifecycle. The items can be product components, end products, product information and knowledge, or engineer tacit knowledge.

(3) Zero, one or many supply chains supply items such as product materials, parts and components, and information/knowledge documents.

(4) Participants can be product knowledge providers and/or product knowledge users.

(5) There are two types of product knowledge sources for each product lifecycle activity, namely, direct knowledge sources and indirect knowledge sources.

If you are using *Word*, use either the Microsoft Equation Editor or the *MathType* add-on (http://www.mathtype.com) for equations in your paper (Insert | Object | Create New | Microsoft Equation *or* MathType Equation). "Float over text" should *not* be selected.

III. Knowledge Integration Framework

Virtual product and process development is a dynamic business process with a recursive, hierarchical structure. To support knowledge service for virtual product and process development, a web service-based framework is proposed using ontological approach to support distributed knowledge service.

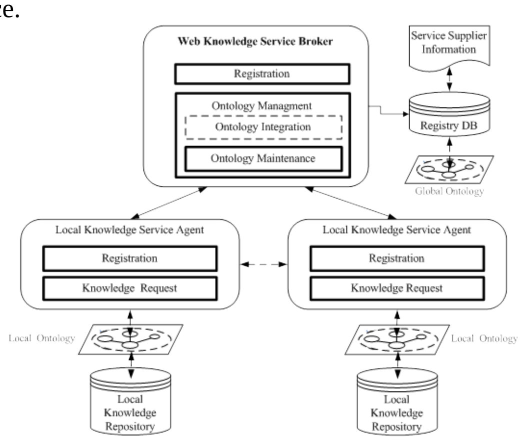

Fig. 2. Web Service-based Knowledge Integration Framework.

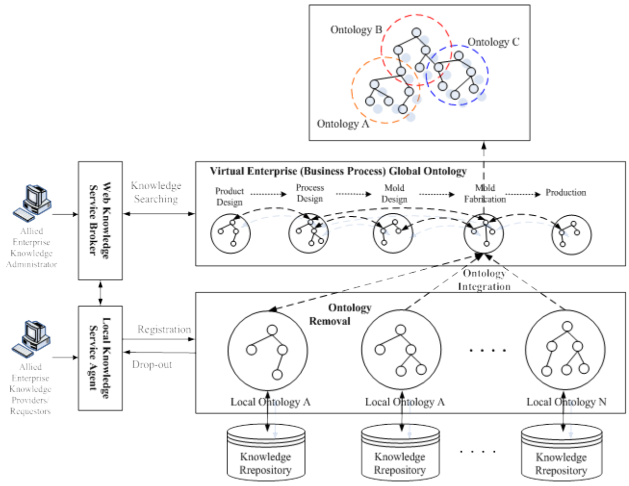

Fig. 3. Ontology Integration Framework.

3.1 Web Service-based Knowledge Integration Framework

A Web Service is a software able to support interoperable machine to machine interaction. Service providers may register their services with public interfaces and bindings defined and described using XML, while service requestors may find appropriate services and then interact with the service provider in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols [13][14][15]. In this proposed web service-based framework, as shown in Fig. 2, each allied enterprise can be a knowledge service provider as well as a knowledge requestor. The framework consists of a web knowledge service broker, and local knowledge service agents. The former which is associated with a registry database containing information of knowledge suppliers and a domain knowledge map is responsible for service registration, and ontology management, while the latter which is associated with a local ontology as the schema of local knowledge repository is instantiated for each allied enterprise to conduct service registration and knowledge request.

3.2 Ontology Integration Framework

As ontology is believed as an effective method for defining entity, property and relationship of knowledge concepts of specific domain [16][17][18][19], ontology is used to define both provided knowledge services and requested knowledge services, as well as the basis for knowledge integration. As depicted in Fig. 3, a global ontology which consists of a set of activity ontology linked with associations is defined to present the domain knowledge map according to the virtual product and process development processes. Each of the activity ontology is the integration of local ontology from allied enterprises involved in this activity. The local ontology itself is the schema of knowledge repository belonging to an allied enterprise.

IV. Knowledge Analysis And Characterization

To provide product knowledge, this section first identifies product knowledge related to product lifecycle, and then categorizes these knowledge into product knowledge categories according to knowledge types. Finally, product knowledge is modeled to facilitate design of a knowledge integration mechanism.

4.1 . Product Knowledge Identification and Classification

The purpose of identifying and analyzing product knowledge is to identify the knowledge associated with a product lifecycle and analyze its characteristics to facilitate the design of product knowledge integration approach. According to the integration definition for process modeling (IDEF0) techniques [20], knowledge involved in each activity in a product lifecycle were identified form its input, output, constraints and references. In addition to the knowledge identified above, people involved in the product lifecycle typically possess tacit knowledge of product knowledge, product domain knowledge and product lifecycle activity knowledge.

The above-mentioned knowledge can be categorized as formal knowledge and practical knowledge. Formal knowledge includes domain concept knowledge, concept knowledge of product lifecycle activities, and theoretical knowledge used when performing product lifecycle activities. Practical knowledge is derived from formal knowledge applied in real-life product lifecycle situations and associated with problem solving.

4.2 Product Knowledge Exploration

The purpose of knowledge exploration is to explore the elements of each area of product knowledge and their relationships.

(1) Formal knowledge exploration:

Concepts are the basis of knowledge and understanding. Therefore, the formal knowledge in this study can be viewed as an aggregation of domain concepts. According to the domain expert cognition, domain core concepts are first defined. These domain core concepts are then categorized and decomposed level by level to establish a primary structure of domain concepts. Subsequently, concepts in this structure are defined for mining related concepts. A domain concept model is thus established gradually using the inside-out approach for mining and exploring concepts and defining the relationships among all concepts.

In this study, product lifecycle knowledge is used as the core for concept exploration. It includes product knowledge and lifecycle knowledge. The former is the aggregation of product components and their relationships, which form the product structure, while the latter contains product lifecycle activities and their sequence, which form a product lifecycle hierarchy. The product structure, product lifecycle hierarchy and their associations together form the basic structure of a domain knowledge map. Other associated concepts are obtained by identifying domain terms in the definition or explanation of concept nodes in the knowledge hierarchy.

(2) Practical knowledge exploration:

Practical knowledge is derived from applying formal knowledge in real-life work situations and problem solving during product and process development. Domain experts, through real-life work, develop various strategies, principles, conventions, and effective practices and experiences based on formal knowledge through personal exposure to, and experiences and observations in, different situations. Practical knowledge can be acquired from product design and process development practices based on real-world development cases. Furthermore, tacit knowledge (including design intent, design know-how, applied theoretical principles, applied heuristics and empirical rules) can be extracted through knowledge elicitation with domain experts and knowledge mining to explain the why and how of product development.

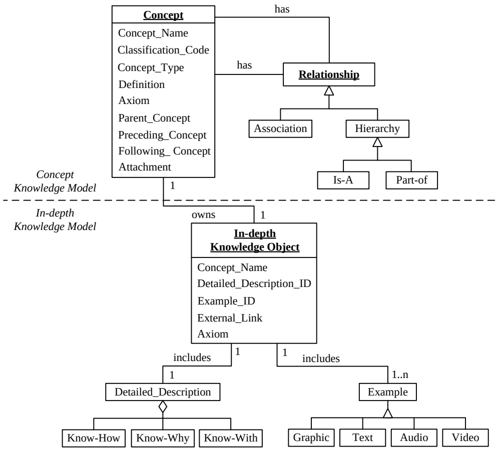

Fig. 4. Schema of formal knowledge model.

4.3. Knowledge Modeling

The aims of knowledge modeling are to analyze the type, characteristics, attributes, and structure of each type of knowledge and to formalize them using modeling techniques and representation schemes.

(1) Formal knowledge modeling:

Developing knowledge in a specific domain starts with the realization of domain core concepts and their relationships. A domain concept can be realized via its definition, features, properties, elements and constraints, as well as through analogy and comparison. A concept model is gradually developed via concept categorization and relationship construction.

In addition to understanding concepts and their relationships, a domain expert has in-depth domain knowledge, including detailed explanations and applications of domain concepts. Therefore, to support product knowledge service on formal knowledge, formal knowledge were further classified into concept knowledge and in-depth knowledge.

Concept knowledge: According to the results of formal knowledge exploration, a concept model, which is a way of presenting knowledge concepts and concept relationships, is developed using ontological engineering technology due to the fact that ontology is considered an effective method for defining an entity or property, and the relationships between knowledge concepts in a specific domain.

The concept model consists of a main structure and associated structures. The former is an aggregation of domain core concepts and their relationships, while each of the latter is the aggregation of concepts identified from definitions of corresponding domain core concepts and their relationships.

As shown in Fig. 4, the main components in the proposed concept model include concepts, relationships and concept objects associated with in-depth knowledge. A concept is defined in terms of a concept name, classification code, concept type, definition, axiom, parent concept, preceding concept, and subsequent concepts. A concept is a tangible or intangible element important to domain applications, and is named using one unique concept name in this concept model.

Additionally, each concept has one classification code, through which concept clusters with similar attributes are categorized into the same class to facilitate the process of concept categorizing, searching and displaying concept relationships or cluster relationships and the drill-down or roll-up activity of category aggregation. A concept can be a core concept or associated concept. A core concept is a concept critical to domain knowledge. A concept definition is the definition or explanation of the concept, from which associated concepts can be identified. Concept axioms serve to model sentences that are always true relative to a concept. The parent concept, preceding concept and following concept define the relationships among concepts. Attachments are the attachments of core concepts, such as in-depth knowledge objects.

Relationships between concepts in the concept model include "Is-A," "Part-of," and "Association." "Is-A" is a hierarchical classification relationship, while "Part-of" is a hierarchical aggregation relationship. Both are the relationships between core concepts. "Association" is a horizontal classification relationship that is connected to relationships between core concepts and associated concepts or relationships among associated concepts.

In this study, Web Ontology Language (OWL) [21], the ontology descriptive language recommended by The World Wide Web Consortium (W3C), was employed to implement the concept knowledge model. The OWL is a novel knowledge representation language based on eXtensible Markup Language (XML) and the Resource Description Framework (RDF), which provides a broad RDF Schema (RDFS) for term definitions and relationship definition for terms, classifications and attributes.

In-depth knowledge: In-depth formal knowledge is defined in terms of objects containing a concept name, detailed description (i.e., operational definition), example, external link and concept axiom, as well as the "know-how," "know-why" and "know-with" of concepts. Concept content can be presented in the form of graphics, text, audio and video representations.

(2) Practical knowledge modeling:
This study uses cases to represent practical knowledge. A case is composed of all activity records for a product lifecycle. A case can be with types of planning, design, development, problem solving, or diagnosis. The content of a case includes explicit knowledge, such as statements of purpose, requirements, strategies and processes, and related engineering data, such as product models, process plans, assembly plans and inspection plans, as well as tacit knowledge such as intentions and explanations of know-how, know-why and know-with associated with explicit knowledge.

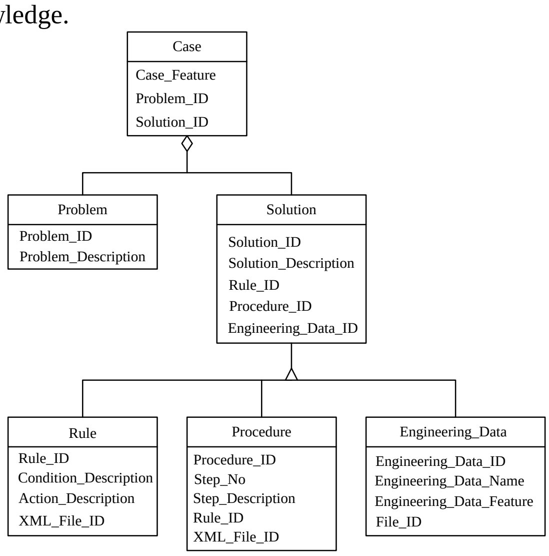

Fig. 5. Schema of case model

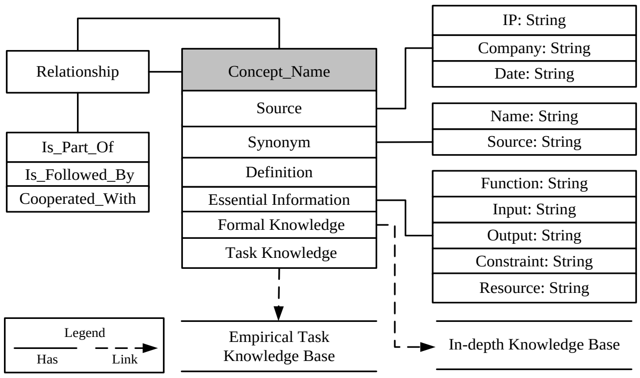

Fig. 6. Local Ontology Schema

To effectively store, organize, manage, and use cases, this study defines a case model as shown in Fig. 5. It contains portions of problem description and solution description. The solution description consists of solution procedure, rule and engineering data.

V. Ontology-Based Knowledge Integration

In this section, an ontology-based integration method is presented to realize product knowledge service.

5.1 Product Lifecycle Ontology Establishment

This research designs a generalized product lifecycle ontology and local ontology schema based on the formal knowledge model discussed in the previous section.

(1) Product Life Cycle Ontology

The product lifecycle ontology encompasses product design, process development, product manufacturing, sales, product in use, post-sale service, and retirement.

(2) Local Ontology Schema

Local ontology schema (Fig. 6) is used by individual enterprises to build their own domain knowledge maps to integrate and share knowledge. The schema is described as follows:

- Concept name: expressing a tacit or explicit knowledge concept name.
- Concept source: recording the name of an enterprise offering knowledge. Whenever an enterprise drops out of the integration mechanism, the link between a concept node and its physical knowledge is removed.
- Concept definition: describing a certain concept such that it is easy to understand and specific. A user can find other relevant concepts in this concept description.
- Synonym: describing the same semantic using different concept terms.
- Essential information: presenting information related to a concept, including function, input, output, constraint, and resource. In this information, the function specifies the actions a concept can perform, whereas a constraint indicates the basic theory or policy for executing a certain task.
- "Resource" are the tools or methods for executing a certain activity or task. Finally, "input" and "output" are the materials required and the results of a certain activity or task execution.
- Formal knowledge: recording the linking address in-depth knowledge that contains detailed descriptive documents for, or examples of, a certain concept.

- Empirical knowledge: recording the linking address that connects specific task knowledge, such as CAD drawings and operational image files.
- Relationship: describing the relationships between concepts using three relationships Is_Part_Of , Is_Followed_By, and Cooperated_With.

5.2 Method for Ontology-based Knowledge Integration

This section presents an ontology integration method based on the above-mentioned knowledge models. The method includes functions of ontology mapping and ontology merging as discussed below.

5.2.1 Ontology mapping

Generally, ontology mapping conducts a matching process to concept names only. Matching results can be "matched name", "partially matched name", or "unmatched name". However, matched names won't be necessarily matched in concepts and unmatched names won't be necessarily unmatched in concepts. Consequently, similarity matching on concept names and similarity matching on essential information and relationship are conducted to identify the similarity between concepts.

The Jaccard Coefficient [22][23] is adopted for ontology mapping and modified slightly for similarity calculation, as detailed below.

5.2.1.1 Name Similarity Matching

Equation (1) is applied for name similarity matching on two concepts. The similarity calculation includes the cross-correlations among Ci^A , Cj^B , Si^A , and Sj^B .

$$\text{Name Sim}\,(Ci^A, Cj^B) = \frac{|Ci^A \cap Cj^B|}{|Ci^A \cup Cj^B|} \quad (1)$$

where Ci^A is the number i concept name term set from ontology A (i.e., $Ci^A = \{Ci_1{}^A, Ci_2{}^A, ..., Ci_n^A\}$), Cj^B is the number j concept name term set from ontology B (i.e., $Cj^B = \{Cj_1{}^B, Cj_2{}^B, ..., Cj_n^B\}$), Si^A is the synonym name term set of number i concept from ontology A (i.e., $Si^A = \{Si_1{}^A, Si_2{}^A, ..., Si_n^A\}$), and Sj^B is the synonym name term set of number j concept from ontology B (i.e., $Sj^B = \{Sj_1{}^B, Sj_2{}^B, ..., Sj_n^B\}$).

In concept name and synonym similarity matching, concept names are first deconstructed as term sets of unit words, the similarity for term sets of two different concept names is then calculated.

5.2.1.2 Essential Information Similarity Matching

To address the issue of term repeatability, the Jaccard Coeffcient formula is included to express the vectors using different weights. The related formulas for essential information similarity matching are as presented in Eqs.(2)~(4).

$$\text{Essential Information}\,(Ci^A, Cj^B) = \frac{|CiE^A \cap CjE^B|}{|CiE^A \cup CjE^B|} \quad (2)$$

$$\text{Jaccard Coefficient} = \frac{\sum_{i=1}^{n} Xi * Yi}{\sum_{i=1}^{n} Xi^2 + \sum_{i=1}^{n} Yi^2 - \sum_{i=1}^{n} Xi * Yi} \quad (3)$$

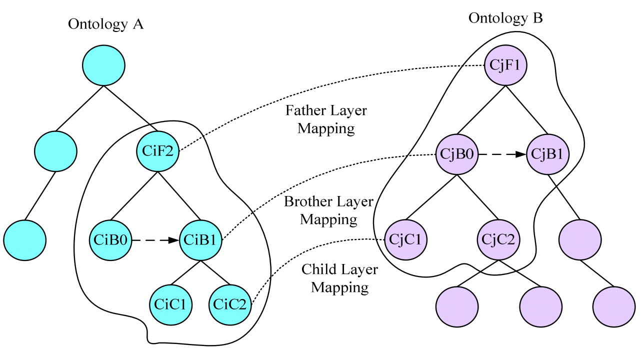

Fig. 7. Relationship Similarity Mapping Diagram

$$\text{Essential Information}\,(Ci^A, Cj^B) = \frac{\sum_{i=1}^{n} CiE^A * CjE^B}{\sum_{i=1}^{n} (CiE^A)^2 + \sum_{i=1}^{n} (CjE^B)^2 - \sum_{i=1}^{n} CiE^A * CjE^B} \quad (4)$$

Both CiE^A and CiE^B are term sets of essential information. In the algorithm, two descriptive term sets from the function of concepts (i.e., $Fi^A = \{Fi1^A, Fi2^A, ..., Fih^A\}$ and $Fj^B = \{Fj1^B, Fj2^B, ..., Fjh^B\}$) are first selected. All descriptive words are then compared to identify the result from vector intersection set value divided by vector union set. Similarly, descriptive words in the order of input, output, constraint and resource from the concept are also compared and calculated. After obtaining all essential information similarities, these similarities are summed to determine an average similarity value for essential information.

5.2.1.3 Relationship Similarity Matching

Equation (5) is applied for relationship similarity matching:

$$\text{Relationship Similarity}\,(Fi, Fj) =$$

$$\frac{1}{3} * \frac{SF}{CiF_x^A + CjF_y^B - SF} + \frac{1}{3} * \frac{SB}{CiB_x^A + CjB_y^B - SB} + \frac{1}{3} * \frac{SC}{CiC_x^A + CjC_y^B - SC} \quad (5)$$

where the $CiFx^A$ is the term set of the father concept for concept A, $CiFx^B$ is the term set of the father concept for concept B, $CiCx^A$ and $CiCx^B$ are the term sets of the child concept for concept A and concept B, respectively, and $CiBx^A$ and $CiBx^B$ are the term sets for process relationship (Fig. 7). Additionally, SF is the number of same father concepts, SB is the number of same brother concepts and SC is the number of same child concepts.

5.2.2 Ontology merging

Ontology merging is employed to deal with the possibility of repeated ontologies when merging concept content and relationships according to ontology mapping results (Fig.8).

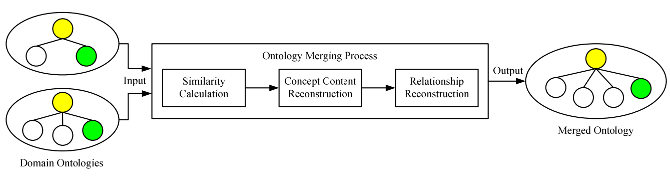

Fig. 8. Ontology Merging Process

ISBN:978-988-18210-2-7

Step 1. Similarity Calculation: The calculation methods are based on the designed ontology mapping algorithm presented in Section 4.2.1.

Step 2. Concept Content Reconstruction: Concept content reconstruction must first consider whether local ontology map with global ontology. If yes, concept content reconstruction and relationship reconstruction are conducted according to mapping results, as follows.

(i) Suppose there is an identical concept in the global ontology: (a) Merging concept content when two or more concept names or synonyms are the same. (b) Merging the concept name into the field of synonyms and their concept content when two or more concepts have the same content.

(ii) Suppose no identical concepts exist in the global ontology: Create a new concept to global ontology that included concept names and content.

Step 3. Relationship Reconstruction: (i) Hierarchical relationship merging: Increasing father concept relationships in the global ontology. (ii) Process relationship merging: Increasing brother concept relationships in the global ontology.

The ontology integration process is designed based on the above results. This process has four sub-processes, namely main process, ontology mapping process, ontology merging process and sub-concept merging process (Fig. 9).

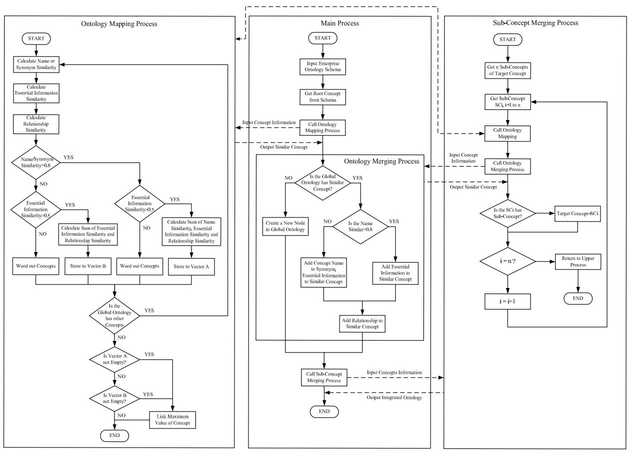

Fig. 9. Ontology Integration Process

I. CONCLUSION

This research develops an ontology-based knowledge integration approach that is able to handle heterogeneity and variation of knowledge form dynamic distributed sources. The outcomes of this study includes: (1) a Web service-based knowledge integration framework, (2) an ontology-based knowledge schema for knowledge integration, and (3) a method for ontology-based knowledge integration. The results of this research may facilitate distributed product and process development through product knowledge integration and sharing.

REFERENCES

[1] Chen, Yuh-Min and Liang, Ming-Wu. "Design and Implementation of a Collaborative Engineering Information System for Allied Concurrent Engineering." *Int. J. of Computer Integrated Manufacturing*. Vol. 13, No. 1, 1999, P. 11-30.

[2] Changchien, S. W., and Lin, M. C., "Design and implementation of a case-based reasoning system for marketing plans", *Expert Systems with Applications*, Vol. 28, No. 1, 2005, pp. 43-53.

[3] Chen, Y. J., Chen, Y. M., Wang, C. B., Chu, H. C., and Tsai, T. N., "Developing a multi-layer reference design retrieval technology for knowledge management in engineering design", *Expert Systems with Applications*, Vol. 29, No. 4, 2005, pp. 839-866.

[4] Chen, Y. J., and Chen, Y. M., "Enabling collaborative product design through distributed engineering knowledge management", *Computers in Industry*, Vol. 59, No. 4, 2008, pp. 395-409.

[5] Ho, C. T., Chen, Y. M., Chen, Y. J., Wang, C. B., "Developing a distributed knowledge model for knowledge management in collaborative development and implementation of an enterprise system," *Robotics and Computer Integrated Manufacturing*, Vol. 20, No. 5, 2004, pp. 439-456.

[6] Aziz, H., Gao, J., Maropoulos, P., and Cheung, W. M., "Open standard, open source and peer-to-peer tools and methods for collaborative product development", *Computers in Industry*, Vol. 56, No. 3, 2005, pp. 260-271.

[7] Jung, J., Choi, I., and Song, M., "An integration architecture for knowledge management systems and business process management systems", *Computers in Industry*, Vol. 58, No. 1, 2007, pp. 21-34.

[8] Ebert, C., and Man, J. D., "Effectively utilizing project, product and process knowledge", *Information and Software Technology*, Vol. 50, No. 6, 2008, pp. 579-594.

[9] Miller, E., "Managing Engineering Data: An Inside Look at PDM." *Computer-Aided Engineering.* July, 1993, SR1 - SR8.

[10] McIntosh, K. G., *Engineering Data Management: A Guide to Successful Implementation*, McGraw-Hill Book Company, New York, 1995.

[11] Bless, P. N., Klabjan, D., Chang, S. Y., "Heuristics for automated knowledge source integration and service composition," *Computers & Operations Research*, 2006.

[12] Bombardier, V., Mazaud, C., Lhoste, P., Vogrig, R., "Contribution of fuzzy reasoning method to knowledge integration in a defect recognition system," *Computers in Industry*, Vol. 58, No. 4, 2007, pp. 355-366.

[13] Daconta, M. C., Obrst, L. J., Smith, K. T., *The semantic web: a guide to the future of XML, web service, and knowledge management*, Wiley Publishing, Inc., United States of America, 2003.

[14] VanderMeer, D., Anindya, D., Kaushik, D., Helen, T., Krithi, R. and Shamkant, B. N., "FUSION: A System Allowing Dynamic Web Service Composition and Automatic Execution", *IEEE International Conference on E-Commerce, Newport Beach, California*, pp.24 - 27, June 2003.

[15] Gómez J. M., Mariano R., Francisco G. and Christoph B., "GODO: Goal driven orchestration for Semantic Web Services", *Proceedings of the WIW 2004 Workshop on WSMO Implementations September*, 2004,pp. 29- 30.

[16] Uschold, M., Gruninger, M., "Ontologies: principles, methods and applications," *Knowledge Sharing and Review*, Vol. 11, No. 2, 1996.

[17] Staab, S., Schnurr H. P., Studer, R., Sure, Y., "Knowledge Process and Ontology," *IEEE Intelligent System*, Vol. 16, No. 1, 2001, pp. 26-34.

[18] Lee, T., Lee, I. H., Lee, S., Lee, S. G., Kim, D., Chun, J., Lee, H., Shim, J., "Building an operational product ontology system," *Electronic Commerce Research and Applications*, Vol. 5, No. 1, 2006, pp. 16-28.

[19] Lin, H. K., Harding, J. A., "A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration," *Computers in Industry*, Vol. 58, No. 5, 2007, pp. 428-437.

[20] Mayer, R. J., Paintec, M. K., and Dewitte, P. S., "IDEF Family of Method for Concurrent Engineering and Business Re-engineering Applications", *Technical Report, Knowledge Based Systems*, 1994.

[21] Huang, N., and Diao, S., "Ontology-based enterprise knowledge integration", *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No 4, 2008, pp. 562-571.

[22] Guha, S., Rastogi, R., Shim, K., "Cure: a efficient clustering algorithm for large database," *ACM SIGNOD International Conference on Management of Data*, 1998.

[23] Kong, H., Hwang, M., Kim, P., "A new methodology for merging the heterogeneous domain ontologies based on the wordnet," *Next Generation Web Service Practices*, 2005.