

# Dynamic Ontology for Supply Chain Information Integration

Wei Yang<sup>1</sup> and Fan Yang<sup>2</sup>

<sup>1</sup>Computer School, Wuhan University, Wuhan 430072, P.R. China  
[william\\_yang2005@163.com](mailto:william_yang2005@163.com)

<sup>2</sup>School of Information Management, Wuhan University, Wuhan 430072, P.R. China  
[yzf\\_613@163.com](mailto:yzf_613@163.com)

**Abstract.** In this paper, ontology technology is applied to Supply Chain information integration, in order to provide unified interface for semantic data and to share, reuse knowledge among information resources. Ontology can solve well the semantic heterogeneity problem of distributed, heterogeneous and autonomous data sources; and the problem of semantic interoperability. Since the objects on Supply Chain are highly dynamic and unpredictable, the information to be integrated is often changing along with the objects, which means that the corresponding ontology also needs to be changed. The process and methods how supply chain dynamic ontology changes is mainly discussed.

**Keywords:** *Supply Chain (SC), Information integration, Dynamic ontology, Global ontology, Local ontology*

## 1. INTRODUCTION

As the development of E-Commerce and information technology, SC becomes more and more agile: the architecture of SC changes rapidly according to the diverse and individual requirements of customers; competition shifts from among enterprises to among SCs; SC management in internal enterprise has extended to the whole industry chain; the resource management has extended too, from internal enterprise (eg. ERP) to external (B2B); the cooperation of enterprises on chain becomes more and more closer--from VMI (Vendor Managed Inventory), JMI (Jointly Managed Inventory) to CPFR (Collaborative Planning, Forecasting). The agility of SC boosts enterprises to constitute Dynamic Alliance, which can integrate information among different enterprises and can easily merge the one that wants to join the alliance into SC information system [1].

Information integration among enterprises on SC is the foundation of the whole SC integration, which means information share among internal and external members on SC. The information includes anything that may affect the behavior of other SC members. Since every enterprise on SC has its own information center, database, operation system, application software and user interface that are always separate from others, many *isolated information islands* emerge. The principal task of SC is to break the isolation and to increase the interoperability of application. It is urgent to integrate these SC data resources that are distributed in physical condition, autonomic

---

Please use the following format when citing this chapter:

Yang, W., Yang, F., 2007, in IFIP International Federation for Information Processing, Volume 254, Research and Practical Issues of Enterprise Information Systems II Volume 1, eds. L. Xu, Tjoa A., Chaudhry S. (Boston: Springer),

in management and heterogeneous in model and to make these SC data can be understand by the computer. To deal with these problems, we introduce the ontology into SC information integration in this paper. Ontology can precisely define knowledge concept and the relationship between them [2], therefore, it can effectively reduce ambiguous understanding, promote information integration and realize information sharing of SC. Ontology provides the guarantee to each node enterprise to communicate conveniently so that improves the operational efficiency of SC enterprises.

There are many researches on ontology building and integration. So far, the main methods of building ontology include TOVE [3], METHONTOLOGY [4] ENTERPRISE [5], KACTUS [6], SENSUS [4], IDEF5 [7], SEVEN STEPS [8] etc. At present, the study on ontology integration has been launched. Reference [9] studied ontology integration from the linguistic point of ontology language, reference [10] studied from the architecture of integration and reference [11] studied from the association of ontology library.

SC is dynamic, so the information integration objects are continuously changing, so is virtual organization ontology. That is, SC ontology is dynamic. In this paper, how to build ontology is not the main concern, but the process and method of how SC dynamic ontology changes are discussed.

## 2. SC DYNAMIC ONTOLOGY

### 2.1 The Origin of SC Dynamic Ontology

The SC is a complex, separate and dynamic network system, of which information increasingly presents its characteristics, such as, high heterogeneity, large quantity, dynamic content and distributed data sources. In view of this point we use ontology to integrate information. SC ontology is dynamic, continuously changing, so we called it dynamic ontology in this paper.

SC is dynamic, which is its main characteristic, because the node enterprises on SC need constantly changing according to business strategy and fluctuation of market requirement.

### 2.2 The Cases of Dynamic Ontology Changing

This paper studies on the process of ontology changing, on condition that original ontology already exists. It emphasizes on the changes of ontology when the internal and external environment have changed. There are cases of environment changes as following:

1. The user has changed the requirement of SC. E.g. some product has been eliminate in the competition, the SC want to produce new product.

2. Technology transforms. E.g. using new technology resulted in SC structure changing.
3. SC node enterprises have changed. E.g. some collaborators quit, and new collaborators enter.
4. The internal structure of SC node enterprise has changed. E.g. a node enterprise wants to add a new department.

In summary, there are two kinds of environment changing: SC structure changing and SC node enterprise changing.

### 2.3 The Content of Dynamic Ontology Changing

According to section 2.2, SC environment changing contains two aspects, so the SC ontology changing from two aspects. The changes of SC structure correspond with the changes of global ontology, e.g. adding a distribution enterprise. The changes of SC members correspond with the change of local ontology, e.g. carrying out personnel restructuring in manufacturing enterprise. Figure 1 shows the SC changing (E. stands for Enterprise), and figure 2 shows the corresponding ontology changing (O. stands for Ontology).

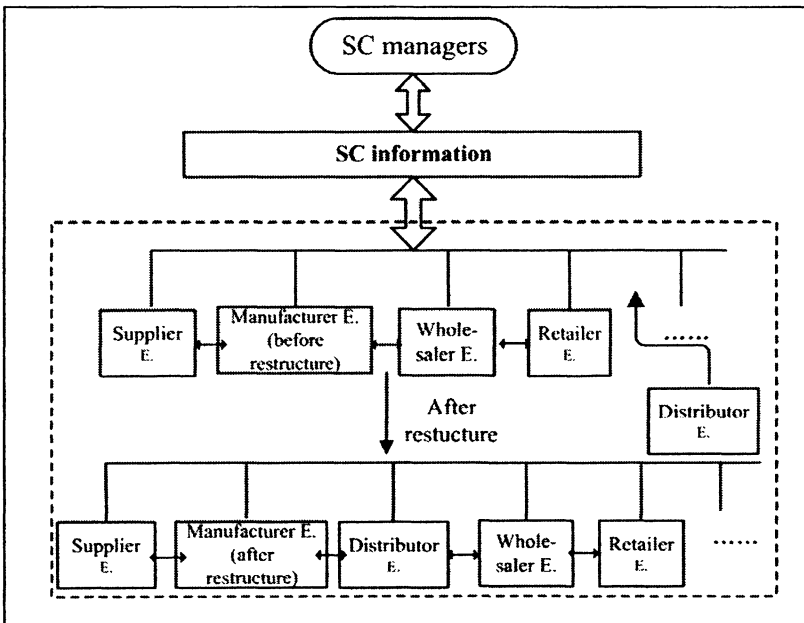


Figure 1. The SC Changing (E. Stands for Enterprise)

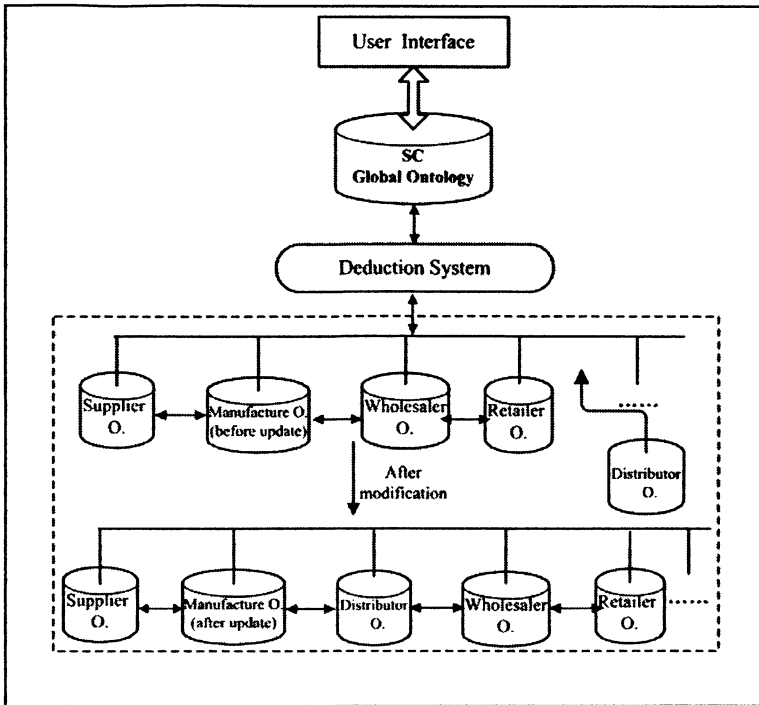


Figure 2. Ontology Changing (O. Stands for Ontology)

### 3. THE METHODS OF DYNAMIC ONTOLOGY CHANGING

According to the analysis in above sections, there are two kinds of SC ontology changing: global ontology and local ontology, each one needs its suited method.

#### 3.1 The Method of SC Global Ontology Changing

The changing of SC global ontology includes adding, deleting and updating each node enterprise (member) on SC. The character of adding is the same as deleting; deleting is the contrary process of adding; so this section mainly discusses the method of adding objects. The method of updating objects will be discussed in the next section about local ontology changing, for a SC ontology object is a local ontology.

Adding an ontology object means integrating a new ontology object into the global ontology. This process uses the theory in PowerLoom deduction system [12] and OTPM [13]. New ontology can be added based on the degree of similarity or

dissimilarity among terms. In this paper, we use similarity to form new global ontology. We take integrating distributor ontology into SC ontology for an example. We only discuss a certain part abstracted from the actual ontology, for the actual one is very complex. The method how to add distributor ontology is illustrated in detail with the following figures.

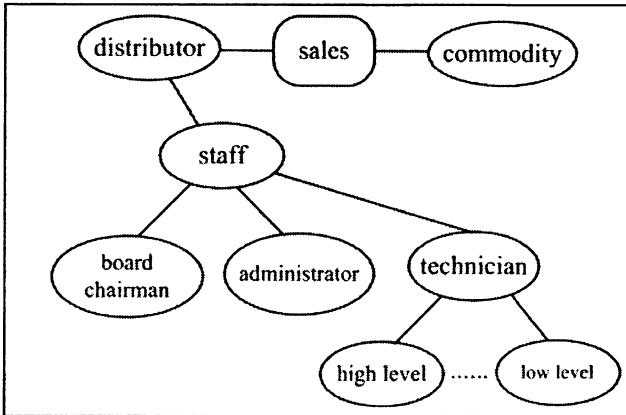


Figure 3. Distributor Ontology P (Abstract from Real Distributor Ontology)

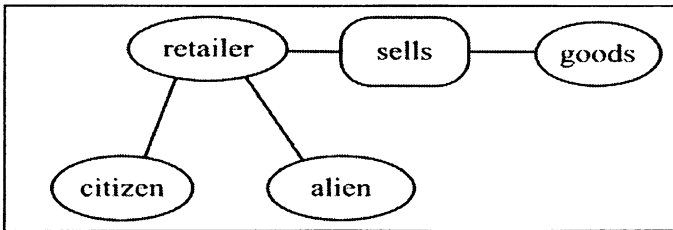


Figure 4. Retailer Ontology Q (Abstract from Real Retailer Ontology)

There are four levels of similarity degree among term connotations: equivalence, specialty, overlap and disjointness. Two terms,  $pt$  and  $qt$ , respectively belongs to distributor ontology  $p$  and retailer ontology  $q$ .  $\phi$  stands for the mapping from the term to its connotation. The relationship between term  $pt$  and  $qt$  are defined as follows:

1. Equivalence.

$pt$  and  $qt$  are equivalent, that is  $pt = qt$ . If and only if their connotation's definitions are the same:

$$(pt=qt) \iff (\phi [pt] = \phi [qt]) \tag{1}$$

For example, *commodity* in Fig.3 distributor ontology and *goods* in Fig.4 retailer ontology, their connotations are the same, so the two terms are equivalent, that is  $commodity = goods$ .

### 2. Specialty.

$pt$  is a specialty of  $qt$ , that is  $pt \leq qt$ . If and only if their connotations' intersection is the same as the term  $pt$ 's connotation:

$$(pt \leq qt) \Leftrightarrow ((\mathcal{C}[pt] \cap \mathcal{C}[qt]) \equiv \mathcal{C}[pt]) \quad (2)$$

For example, *staff* in Fig.3 is a specialty of *distributor* in Fig.4, that is  $staff \leq distributor$ .

### 3. Overlap.

$pt$  and  $qt$  overlap, that is  $pt \cap qt$ , if and only if their connotations' intersection is not null (connotations' intersection concept  $T$  is always true).

$$(pt \cap qt) \Leftrightarrow (((\mathcal{C}[pt] \cap \mathcal{C}[qt]) \equiv \mathcal{C}[T]) \wedge \neg \mathcal{C}[T] \equiv \text{False}) \quad (3)$$

For example, *staff* in Fig.3 and *alien* in Fig.4 overlap, that is  $staff \cap alien$ .

### 4. Disjointness.

$pt$  and  $qt$  disjoint. If and only if their connotations' intersection is null:

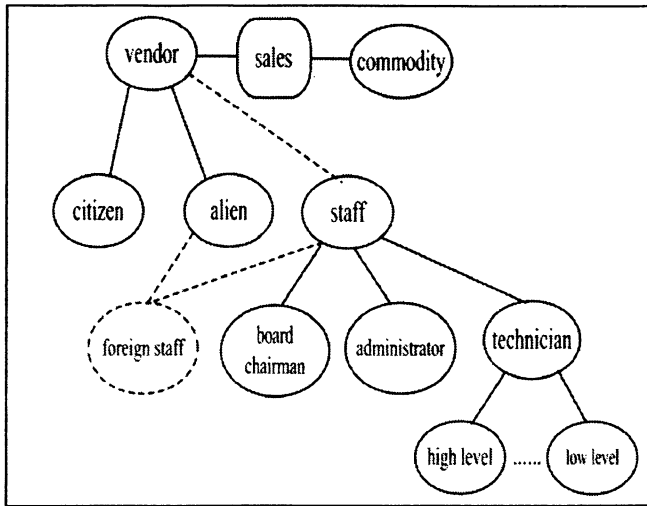
$$\neg (pt \cap qt) \Leftrightarrow ((\mathcal{C}[pt] \cap \mathcal{C}[qt]) = \Phi) \quad (\Phi \text{ stands for null}) \quad (4)$$

For example, *staff* in Fig.3 and *goods* in Fig.4 are two disjoint terms.

There are several integration regulations, too. Disjointness doesn't matter much with integration, so we don't discuss this case in this paper.

5. **Regulation 1.** Equivalence integration. If two terms' definition D1 and D2 are equivalent, we can combine these two definitions into one (D1 or D2). E.g.  $commodity = goods$ , so we can use *commodity* only instead of *commodity* and *goods*.
6. **Regulation 2.** Specialty integration. If definition D1 is a specialty of definition D2, we can build the special relationship between them. E.g.  $staff \leq distributor$ .
7. **Regulation 3.** Overlap integration. If definition D1 and D2 overlap, we can create 3rd definition D3 to stand for their intersection. D3 can have its instance or not, it depends on the experts. E.g.  $staff \cap alien$ , we use new definition *foreign staff* instead of their intersection. Foreign staff has no instances.

According to the above regulations, we join distributor ontology (shown in Fig.3) and retailer ontology in SC global ontology (shown in Fig.4) into a new SC global ontology (shown in Fig.5).



**Figure 5. The Result of Adding Distributor Ontology into SC Global Ontology (Abstract from the Real One)**

### 3.2 The Method of SC Local Ontology Changing

Local ontology means certain domain ontology, such as manufacture ontology and distributor ontology etc. We can also call it member ontology. At present, this topic attracts many attentions. There are several ways to call ontology changing, for example, ontology evolving and ontology updating etc. But they are the same in nature; all of them modify ontology concepts and the relationship among them, in order to suit the changing environment. M. Klein and D. Fensel bring forward that ontology changing contains three aspects: changing in domain, changing in concept, changing in regulation [14]. Compared to SC global ontology changing, local ontology changing is more detailed in recent researches.

There are many methods of local ontology changing [15, 16]. We use *OWL-Oriented Evolution of the Ontology*, referring to [17]. This method describes the relationship among concept, slot and side. It achieves the Related-Concept set of changing objects through *Get-Related-concept* algorithm, and updates through *Change-Related-Concept* algorithm. In view of SC, changing should place mainly on concept, instance and application of the ontology that needed modification. E.g. to carry out personnel restructuring in manufacture enterprise, we should change concepts and corresponding relationships involving in personnel.

#### 4. THE CHANGING PROCESS MODEL OF SC DYNAMIC ONTOLOGY

In the above sections, we introduce the reason, content and methods of SC dynamic ontology changing. To sum up, we put forward the changing process model of SC dynamic ontology. This model also consists of two kinds of updating: SC global ontology and node enterprise (member) ontology, as shown in Fig.6.

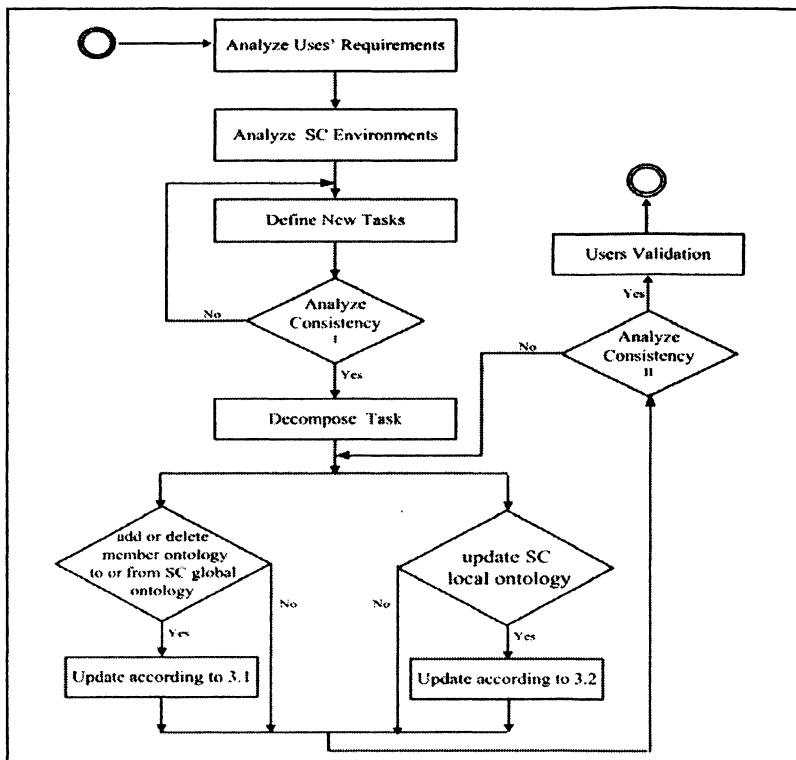


Figure 6. The Changing Process Model of SC Dynamic Ontology

In figure 6, at first, the users ask for changing ontology. Then, the ontology managers analyze the requirements of users and the internal and external environment of SC. Subsequently managers define ontology changing task. If the task is complex enough, it should be disassembled. If global ontology of SC needs changing, it could be changed following the method discussed in section 3.1; else if local ontology of node enterprise needs changing, it could be changed following the method mentioned in section 3.2. Finally, the users check the updating results. If users validate, SC



dynamic ontology changing will finish. If there are other requirements, the next updating cycle will begin.

*Consistency analysis*, which aims to reduce conflicts, appears twice in the process model. It deals with semantic conflicts among ontology concepts, data error of instances and function error of applications caused by ontology changing. But the tasks of the two kinds of consistency analysis are not the same with each other. The former one occurs before the implementation of changing. It analyzes all kinds of conflicts, which may be occurred while in updating process of SC system, to make sure the feasibility of ontology changing. The latter one occurs after the updating tasks. Its task is to ensure the consistency of the whole SC system.

In this paper, the changing of dynamic ontology is emphasized, so the way in which ontology changes is mainly described in the process model. Other topics of dynamic ontology, such as bulletin management and version management, are not mentioned.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, the research background is SC, and we mainly study its dynamic ontology. Through the example of adding distributor ontology to SC global ontology and implementing personnel restructuring in manufacture department, we analyzed the origin, reason and content of dynamic ontology changing, brought forward the corresponding method and suggested the dynamic ontology changing process model. All changing issues focus on two aspects: global ontology of SC and local ontology of node enterprise.

In this paper, the examples abstracted from the actual ontology are very simple, but the actual ones are much more complex and can change at different granularity [16]. Therefore, we should pay more attention to the popularization of content, process and method discussed in this paper, and we will study the applicability of SC dynamic ontology changing regulations in future.

## REFERENCES

1. D. Fensel and C. Bussler, The Web Service Modeling Framework WSMF, *Electronic Commerce Research and Applications*. Volume 1, Number2, pp.113-137, (2002).
2. T.R. Gruber, *A translation approach to portable ontology specifications*, Tech Rep: Logic-92-1, Stanford University (1993).
3. M. Gruninger, Designing and Evaluating Generic Ontologies, in *Proc. of ECAI96's Workshop on Ontological Engineering* (1996), pp.53-64.
4. M. Fernández, Overview of Methodologies for Building Ontologies, in *Proc. of IJCAI99's Workshop in Ontologies and Problem Solving Methods: Lessons Learned and Future Trends* (1999).
5. J. Stader, Results of the Enterprise Project, in *Proc. of the 16th Int. conference of the British Computer Society Specialist Group on Expert Systems* (Cambridge, UK, 1996).

6. N. F. Noy, S. Kunnatur, M. Klein, and M.A. Musen, Tracking Changes During Ontology Evolution, in *Proc. of the 3rd International Semantic Web Conference (ISWC-04)*, (2004).
7. KBSI. IDEF5 Ontology Description Capture Overview. <http://www.idef.com/idef5.html> (Accessed October 1, 2003).
8. Natalya F. Noy and Deborah L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology* (Auegst, 2001). [http://protege.Stanford.edu/publications/otology\\_development/ontolog101.pdf](http://protege.Stanford.edu/publications/otology_development/ontolog101.pdf) (Accessed October 12, 2002).
9. M. Klein, Combining and Relating Ontologies: An Analysis of Problems and Solutions, in *Proc. Of IJCAI'01 Workshop on Ontologies and Information Sharing* (Seattle, WA, 2001), pp.53-62.
10. H. Wache and T. Vgele, Ontology-based Integration of Information—a survey of Existing Approaches, in *Proc. of the Workshop Ontologies and Information Sharing*, (2001).
11. M. Prasenjit, A Graph-oriented Model for Articulation of Ontology Interdependencies, in *Proc. Of EDBT'00* (Springer: Verlag, 2000), pp.86-100.
12. M.R. MacGregor, H. Chalupsky, and E.R. Melz, *PowerLoom Manual*, University of Southern California (November, 1997). <http://www.isi.edu/isd/LOOM/PowerLoom/documentation/manul.pdf>.
13. Y. Zhang, H. Zhang, and W. Zhang, Ontology-Based Information Integration Mapping Construction Between Global View and Local Views in Virtual Organization, *MINI-MICRO SYSTEMS*. Volume 3, Number 27, pp.253-257, (2006).
14. M. Klein and D. Fensel, Ontology Versioning on the Semantic Web, in *Proceedings of the International Semantic Web Working Symposium* (2001), pp.75-91.
15. G. Elouris, D. Plexousakis, and G. Antoniou, Evolving Ontology Evolution, in *SOFSEM 2006, LNCS 3831*, eds. J.Wiedermannetal (Springer: Verlag Berlin Heidelberg, 2006), pp.14-29.
16. L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic, User-Driven Ontology Evolution Mangement, in *EKAW 2002, LNAI 2473*, eds. A. Gómez-Pérez and V.R. Benjamins, (Springer: Verlag Berlin Heidelberg, 2002), pp.285-300..
17. M. Zhou, J. Gao, and F. Li, OWL-Oriented Evolution of the Ontology, *Journal of Computer-aided design & computer graphics*. Volume 3, Number 17, pp.584-591, (2005).