

Supply chain teaming: A knowledge intensive approach for finding and selecting partners

Yu-Liang Chi

Dept. of Information Management, Chung Yuan Christian University,
200 Chung-Pei Rd., Chung-Li 32023, Taiwan, R.O.C.
maxchi@cycu.edu.tw

Abstract. This study shows how the semantic rules in conjunction with ontologies can be used for managing the intricate network among supply chains partners. Nowadays, an enterprise is no longer a self-sufficient firm but an integral part of supply chain. When structuring a supply chain network, it is problematic to identify who the partners are across multiple supply chain tiers. Worst, an inclusion of potential partners may cause the links to become an intricate network. We introduce ontological knowledge framework which first building concepts for defining enterprises and products, describing instances with properties as facts, and developing semantic rules to infer inter- and intra relationships among facts. The major advantage of this design is its ease to implement that individual firm and its parts manage their direct and known knowledge. Small examples were presented using ontology and semantic rules.

Keywords: Supply chain, Teaming; Knowledge-based system; Ontology; Rules

1 Introduction

Modern business competition is no longer individual company versus another company but two opposite supply chains against each other [13]. Most commercial brands of products today consist of collaborative efforts from various suppliers rather than efforts of a firm alone [11]. Accordingly, supply chain is a business paradigm for present and future. Modern business encounters great difficulties of dynamic environment more than ever. A successful supply chain management (SCM) can help achieve agility through the ability to respond quickly to customer demand and by reducing operating costs. Most firms employ either commercial or customized SCM tools to make their business successful. The basic requirements of supply chain partners are willing to accommodate the uncertainties and variations in each other's businesses. Thus, SCM is a typical way of managing the business and its relationships.

Traditional supply chain focused on enhancing efficiency of business functions. However, modern business encounters dynamic products and complicated supply-demand relationships more than ever. At the strategic manufacturing planning level, capable firms must not only control and collaborate with existing supply chain partners efficiently but also need to keep watch for potential partners [14]. Since the upstream or downstream of supply chain can be extended one tier to another,

monitoring all tiers of suppliers is not easy [2]. A strategic planning of supply chain partners is a company's vision of what it wants to cope with changes in competitive environments [12]. Supply chain partners in both up- and downstream might be extended many tiers. In addition, any partner in different tier can derive its own partners that cause multi-dimensions rather than a single hierarchy of supply chains. The challenge in controlling such a network stems from the nature of relationships between supply chain partners [17]. Though many companies have utilized supply chain management tools, the challenge then arise how many tiers of suppliers that a firm can really handle them. Olhager and Selldin have reported a SCM utilization survey of Swedish manufacturing firms [16]. There is a low rate of results concerning supply chain integration especially in integrating upstream operations such as 1st and 2nd tier suppliers.

As mentioned above, identifying entire supply chain partnerships is one of key success elements in modern manufacturing collaboration. Without knowing which partners of a supply chain, planning system cannot establish specific supply chain visions. Absence of knowing potential partners, we cannot make an agile composition of supply chain partners. To extend supply chain planning and control, the main purposes of this study are threefold: (1) How to identify who the partners of the entire supply chain are. (2) How to implement network traversal; and (3) How to provide mechanism in managing a composition of supply chain partners. This study first utilized ontological knowledge framework to model the task domain related to supply chain partners. Then, we developed rules to define how supply chain partners are behaved, how a production planning is made, and so on. After knowledge and rules have been built, existing facts can be linked to generate and infer proper connection of a spanned network.

2 Building supply chain partners profiles using ontology

Issues in making a successful supply chain can be various and complicated. This study is interested in structuring supply chain networks. That is, who are the members with whom to link processes? Since a wide spectrum of the supply chain partners can be extended as an intricate network, connecting suppliers across multiple tiers of entire supply chain is a challenge task. Increasingly, the efforts of constructing the supply chain include model building, facts and relationships collection, and lots of domain knowledge accumulation. In order to better manage supply chain members, knowledge-intensive approaches such as ontology is suited to the occasions. Several ontology engineering approaches and implementations have been discussed in studies [5][10].

Ontology provides an explicit specification of a conceptualization to express shared human perspectives of the real world [1][4]. Additionally, a concept is an abstract, simplified world view used for representational purpose [6]. Like most knowledge-intensive approaches, building ontology is known as knowledge engineering that generally includes several successive processes such as knowledge acquisition, modeling, and representation. Accordingly, the major task of building ontology is translating goal-oriented or problem-solving activities into a

systematically knowledge needed to solve a problem. As [18] point out that knowledge required to solve problems is affected by the nature of problem. To achieve clear definitions of common understanding, knowledge development relies on the integration of perspectives from task domains, communities, and applications.

In order to elucidate the principles above, this study assumes partnerships among supply chain members are based on supply-demand relations of products or services. We first determine the scope of the task domain and the vocabulary necessary to describe the conceptual structures. The preparation stage of knowledge acquisition was implemented on collecting relevant concepts and properties of supply chain members. As shown in Table 1, basic concepts of supply chain such as *Company* and *Product* are identified. A listing of properties beyond each concept is used to describe concept's characteristics. Regarding the value inside properties, this study informally divided properties into two categories: "Asserted Properties" and "Inferred Properties". Asserted properties are allowing input known facts or called explicit knowledge. For example, company's *Location* and *Capital* are known facts. Inferred properties are unapparent facts or called implicit knowledge that leads an inference using known facts or assumptions. For example, *Potential_Suppliers* and *Potential_Customers* can be inferred by known facts.

Table 1. Properties of the supply chain model

	Company	Product	Product_Type	Country	..
Asserted Properties	Location Country Capital CEO GM employeeAmount hasProducts	Factory hasType unitPrice CapacityMonth needParts Capacity_Utilization Customers	standardNo Descriptions	Capital_city Airports Harbors Tax_rate hasArea	
Inferred Properties	Suppliers Customers Potential_Suppliers Potential_Customers	Suppliers Potential_Suppliers Potential_Customers Available_Capacity Plan_Suppliers	hasFactory		..

To select notation or formalism used for representing the knowledge to be stored in ontology, OWL (Ontology Web Language) is utilized in this study. OWL is being developed by the W3C Web ontology working group. OWL is primarily designed to represent information about categories of objects and how objects are interrelated [7]. OWL ontology consists of classes, properties, and individuals, which roughly correspond to concepts, roles, and instances of ontology. OWL classes are interpreted as sets that contain similar individuals. OWL properties represent relationships between two individuals. Two types of properties are object properties and datatype properties. Object properties link an individual to an individual. Datatype properties link an individual to an XML schema datatype. Individuals represent objects in the domain that we are interested in. More leisurely descriptions can be found in OWL specification.

3 Finding semantic relationships using semantic rules

OWL ontology is designed to represent information about concepts of instances and how instances are interrelated. Description Logics (DL) further helps concepts to formally define restrictions that constrained instances' behaviors. Consequently, OWL ontology with DL facilitated knowledge model in an abstract view. In this study, however, an individual firm in a supply chain may have multiple roles such as a supplier, a customer or both roles. The roles of a firm are dynamic depending upon the needs of its products (i.e., OWL instances) in supply-demand relationships. Therefore, semantic relationships of individuals are usually described by its properties rather than its concept. Horrocks et al. have reported several DL inference limitations in instance properties such as syntactic, expressive, and computation [8]. One critical issue of DL is that it is incapable of represent rules in DL engines that would lead to the undecidability of inference problem. The Semantic Web Rule Language (SWRL) is an emerging technology developed to address above difficulties of OWL and DL [3]. SWRL extends the set of OWL axioms to include Horn-like rules that facilitates SWRL rules to be combined with an OWL knowledge base [9]. The SWRL rules are written as antecedent-consequent pairs. Both antecedent (or rule body) and consequent (or rule head) are conjunctions of one or more atoms written $atom_1 \wedge \dots \wedge atom_n$, variables are indicated by a question mark (e.g., $?x$). For example, a simple supply chain is about suppliers and relationships (e.g., *hasSupplier*, *hasCompany*), etc. If we want to assert a rule to find the 2nd tier's suppliers, the rule would then be:

$$\text{hasSupplier}(?x, ?y) \wedge \text{hasSupplier}(?y, ?z) \rightarrow \text{has_2}^{\text{nd}}\text{_tier_supplier}(?x, ?z)$$

Implementing this rule would take variant individuals to indicate corresponding variables. For example, the following particular individuals C_1 , C_2 , C_3 indicates variables x , y , z respectively. After inference computing, individual C_1 obtains a 2nd tier's supplier C_3 . SWRL also supports built-ins predicates, which expand its expressive power such as comparisons and math operations. The prefix "swrlb:" is used by convention to denote the SWRL built-ins namespace. For example, *swrlb:subtract(?a, ?b, ?b)* is math built-ins. The SWRL rules can be edited in the Protégé OWL platform by choosing the plugin named SWRLTab. The rule engine such as JESS (Java Expert System Shell) needs to connect in the Protégé platform before execution. The JESS software consists of three components including a rule base, a fact base and an execution engine [15]. Fig. 1 shows the protégé embedded JESS, two frames from top to bottom are a SWRL rules editor and a JESS runtime interface. The rules editor allows users to write rules as text. The JESS runtime interface provides functions to perform the JESS Rule engine. The rectangle drawn by dashed lines in the bottom contains three buttons that perform: translating OWL ontologies and SWRL rules as JESS facts and rules, fire the rule engine, and write new facts back to the OWL, respectively.

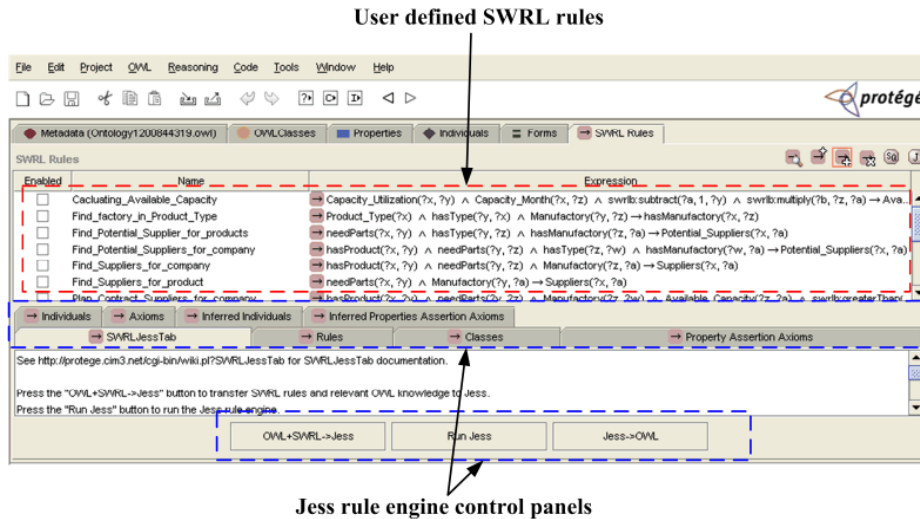


Fig. 1. A screenshot of editing SWRL rules and JESS connections

The SWRL rules editor plug-in is a Java-based API, called SWRL Factory, which allows developers to access SWRL rules in OWL ontologies. On the other hand, JESS rules engine also provides a Java-based API, called SWRL-JESS Bridge, to execute its rule engine. For applications development needs, developers can use both APIs to develop runtime interactions programmatically.

4 Examples of accessing supply chain networks

This study has experimented with OWL ontology on building supply chain partners knowledge. Three examples of using SWRL rules are shown below in finding multiple tiers' suppliers, potential suppliers, and a production plan based on available capacities of products. The representation of SWRL rules is relatively straightforward. We have noted that there are several possible ways to reach same rules functions of our examples. Once the rules have been represented, the JESS engine can perform rules inference. As rules fire, OWL asserted facts, inferred facts performed by the DL engine, and SWRL rule facts can be inserted into the fact base.

Example 1: Finding suppliers across multiple partner tiers.

An Individual firm usually has various products that are supported by different partners. To find all suppliers across multiple supply chain tiers, SWRL rules are utilized to explore relationships starting from interpreting products rather than suppliers. For example, the following SWRL rule would be to assert that the combination of *hasProducts*, *needParts* and *Factory* properties implies the *Suppliers* property.

$$\text{hasProducts}(\text{?x}, \text{?y}) \wedge \text{needParts}(\text{?y}, \text{?z}) \wedge \text{Factory}(\text{?z}, \text{?a}) \rightarrow \text{Suppliers}(\text{?x}, \text{?a})$$

The above rule provided only a general syntax for finding next suppliers. Users can iteratively perform the rule to find the furthest upstream partners of the supply chain. Fig. 2 contained I , II and III blocks to demonstrate successive iterations. Inside each block, a shaded rectangle represents a company, a circle represents single product, small shaded circles represent properties of a class, and arrow lines denote inference progression. The facts data can be inserted into variables (e.g. ?x). From this rule in the first block, if the C_0 company has P_0 product, P_0 need P_1 parts, and P_1 is manufactured by C_1 company then C_0 company get a supplier is C_1 company. The second block performs the same rule but giving different facts. The rule inference result is from C_1 company and P_1 product to infer that C_1 company has a supplier C_2 company. Table 2 listed the iterations step by step. Through properties conjunctions of the SWRL rules, users can possibly obtain suppliers across entire supply chains.

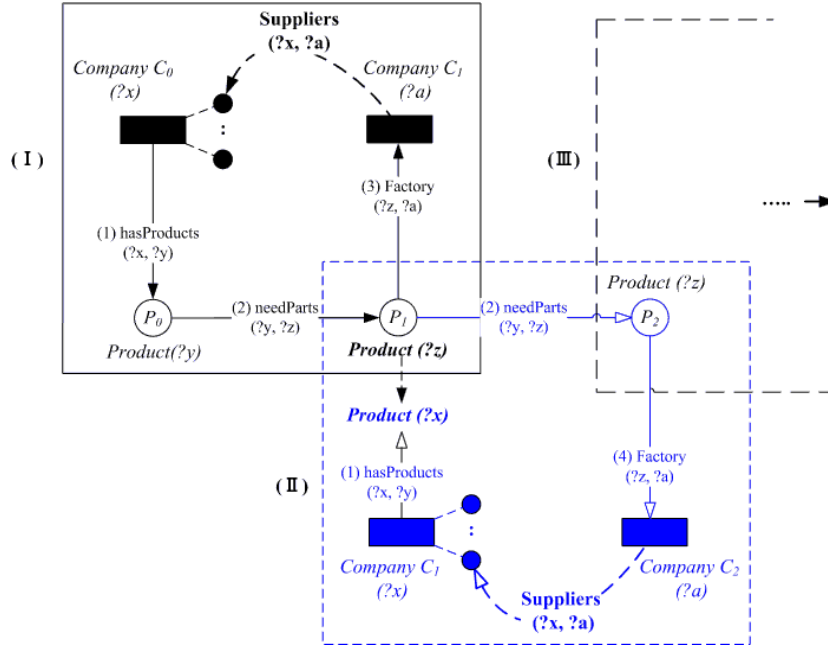


Fig. 2. A rule decomposition steps for finding a company's suppliers

Table 2. Iterations of using SWRL rules to infer suppliers

Iteration	hasProducts(?x, ?y)	needParts(?y, ?z)	Factory(?z, ?a)	Supplier(?x, ?a)
1	(C_0, P_0)	(P_0, P_1)	(P_1, C_1)	(C_0, C_1)
2	(C_1, P_1)	(P_1, P_2)	(P_2, C_2)	(C_1, C_2)
3	(C_2, P_2)	(P_2, P_3)	(P_3, C_3)	(C_2, C_3)

Example 2: Finding potential suppliers across multiple partner tiers.

The Previous example has shown how to identify contract suppliers of a firm. In this example, we try to find potential suppliers. The potential suppliers are simply defined that they are capable to provide products with same functions to satisfy the needs of a customer. To identify potential suppliers, SWRL rules first identify the product type of existing supplier and then using the product type to seek suppliers which manufacture same type of products comparing with existing contract suppliers. The following SWRL rule asserts that the conjunction of *hasProducts*, *needParts*, *Factory*, *hasType* and *hasFactory* properties can be used to conclude who are the potential suppliers and assign company names into the *Potential_Suppliers* property.

```
hasProducts(?x, ?y) ^ needParts(?y, ?z) ^
Factory(?z, ?a) ^ hasType(?z, ?w) ^ hasFactory(?w, ?b)
→ Potential_Suppliers(?x, ?b)
```

Fig. 3 demonstrated continuative inference processes of identifying potential suppliers. From this rule, if C_0 company has P_0 product, P_0 needs P_1 parts, P_1 is a kind of T_1 type, and T_1 can be manufactured by $C_{a..}$ company then C_0 company has potential suppliers $C_{a..}$.

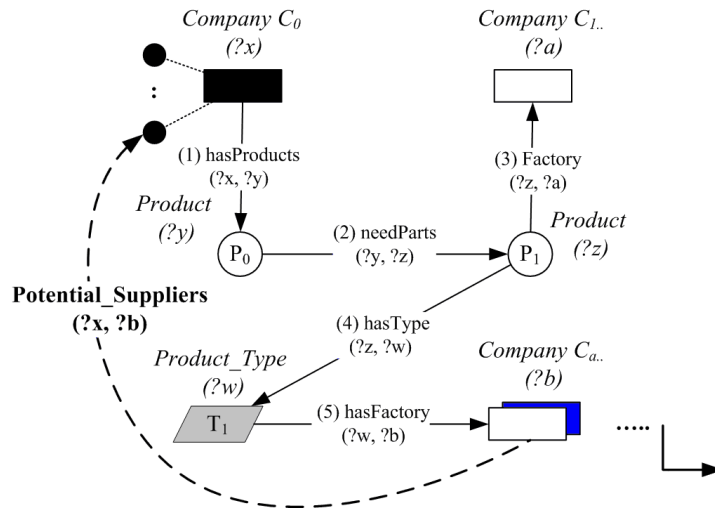


Fig. 3. A rule decomposition steps for finding a firm's potential suppliers

Example 3: A production forecasting based on available capacities of products

The last example about production forecasting is much more complex. Assuming a firm plans to raise 20% production of a product. If contract suppliers have available capacity, they have a high priority to join the new plan. If contract suppliers are short of production capacities then potential suppliers become new suppliers of the plan. To simplify this scenario, two assumptions are made: (1) each company has reported

accurate capacity utilization; (2) a product and each composing part are a 1-to-1 relationship. Three SWRL rules are given to implement the planning goal.

- A rule of calculating available capacity of each product

The following SWRL rule utilizes the values of product's asserted properties and some SWRL math built-ins functions to calculate available capacity of each product. The *swrlb:subtract(?a, 1, ?y)* requires three arguments that the first argument is the arithmetic difference of the second argument minus the third argument. The *swrlb:multiply(?b, ?z, ?a)* also requires three arguments that the first argument is the arithmetic product of the second argument multiplies the third argument. From this rule, if P_0 product has an asserted capacity utilization (i.e. $?y$; say 80%), the value of available capacity utilization will assign 0.2 to $?a$ (e.g. $1-0.8=0.2$), P_0 has an asserted monthly production capacity (i.e. $?z$; say 20000), the amount of available capacity will assign 4000 to $?b$ (e.g. $20000*0.2=4000$) then the value will be inserted into the *Available_Capacity* property of the product.

```
Capacity_utilization(?x, ?y) ^ swrlb:subtract(?a,
1, ?y) ^ capacityMonth(?x, ?z) ^
swrlb:multiply(?b, ?z, ?a) → Available_Capacity(?x, ?b)
```

- A rule of selecting qualified contract suppliers

The following SWRL rule selects qualified contract partners by verifying that the available capacity of a product is greater than a value. Assuming a firm decides to increase 2000 units of a product. From this rule, if C_0 company has P_0 product, P_0 needs P_1 parts, P_1 is manufactured by C_1 company, obtaining the value of the *Available_Capacity* that we have calculated, and using a SWRL math built-ins function *swrlb:greaterThan(?a, 2000)* to compare with the number 2000 then the qualified suppliers are inserted into the *Plan_Suppliers* property of the product. Fig. 4 demonstrated the identifying process step by step.

```
hasProducts(?x, ?y) ^ needParts(?y, ?z) ^ Factory (?z, ?w) ^
Available_capacity(?z, ?a) ^ swrlb:greaterThan(?a, 2000)
→ Plan_Suppliers(?x, ?w)
```

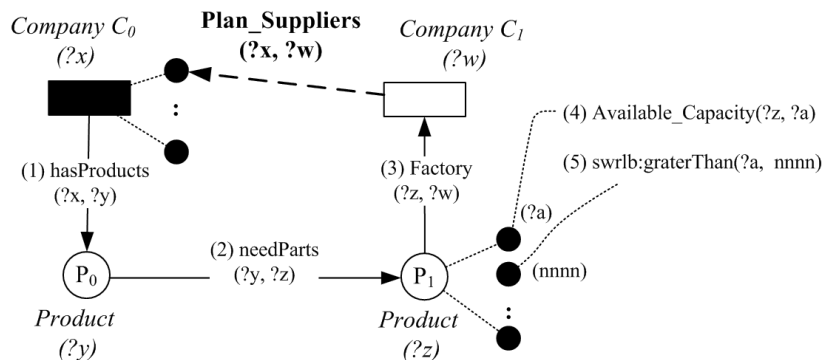


Fig. 4. A rule decomposition steps for selecting qualified contract suppliers

- A rule for selecting qualified potential suppliers.

If contract suppliers are unavailable to support the capacity needs of the new plan, a firm has to seek qualified potential partners. The criterion of selecting partners is verifying available capacity of a product is greater than the set value. From the following rule, if C_0 company has P_0 product, P_0 looking for potential suppliers from *Potential_Suppliers* property of the product, obtaining the products from potential suppliers, and using comparing process to confirm the available capacity of a product great than 2000 then the supplier will be inserted into the *Plan_Suppliers* property. Fig. 5 demonstrated the identifying process step by step.

```
hasProduct(?x, ?y) ^ Potential_Suppliers(?y, ?z)
^ hasProduct(?z, ?a) ^ Available_Capacity(?a, ?b) ^
swrlb:greaterThan(?b, 2000) → Plan_Suppliers(?x, ?z)
```

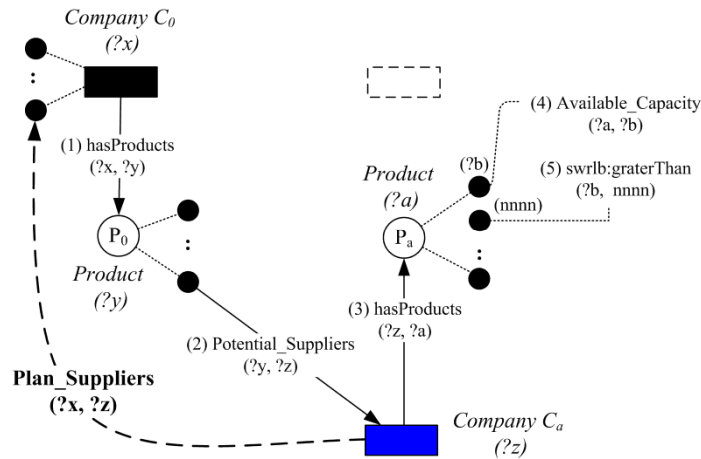


Fig. 5. A rule decomposition steps for selecting qualified potential suppliers

5 Conclusion

A successful SCM requires constructing close relationships with suppliers and customers in order to survive in the highly competitive market. Since supply chain is a network of multiple businesses and relationships, monitored activities of business partners across multiple tiers of a supply chain is challenges. To address complicated relationships among partners, this study introduced an ontological approach to model knowledge about specific tasks. Based on knowledge engineering principles, this study identified necessary concepts, characteristics, and relationships among supply chain members. Then, the Protégé OWL tool is utilized to edit this knowledge into

OWL-based ontologies. Since OWL ontology mainly provides an abstract view in the concept level, semantic relationships among individuals are difficult to discover. SWRL rules provide procedural knowledge power to enhance limitations of ontology inference especially in discovering semantic relationships among instances. This study then utilized SWRL rules in conjunction with ontologies to manage the intricate supply chain network. Three examples were provided including finding contract suppliers multiple across multiple tiers of supply chains, finding potential suppliers, and a production plan based on available capacities of products.

Since the knowledge of supply chain partners have been built, more applications can be further created to infer new and implicit knowledge. Based on our experiences with building SWRL rules and OWL ontologies we conclude that such approaches will be crucial to achieve the following advantages:

- Developers has captured and modeled knowledge of the task domain into ontologies that the computer has all the knowledge needed to solve specific problems.
- Information holders only need to edit or update their known facts as ontological knowledge. The separation of knowledge framework and information collection makes easier for both developers and users.
- The rules are solid expertise of common agreements that describe how things get done. Existing facts obey the guidance of predefined rules. If facts change, new connections of supply chain networks can be rapidly rebuilt by inference services.

Although this study presented only a few simple examples of tracing suppliers, the ontology is scalable by including more factual individuals, enterprises, products and product types. Further, by following the above development processes, both domain ontology and task ontology can be expanded to solve more complex and practical problems. A Web-based system can be developed by simply utilizing Java-based applications supported by Jena API, Protégé API and SWRL API. Consequently, an inferred ontology can be considered a knowledge base with intelligence.

Acknowledgment

The author would like to thank the National Science Council (NSC) of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSC 96-2416-H-033-002-MY3.

References

1. Chi, Y.-L.: Elicitation Synergy of Extracting Conceptual Tags and Hierarchies in Textual Document. *Expert Systems with Applications*, 32, 349--357 (2007)
2. Dudek, G., Stadler, H.: Negotiation-based Collaborative Planning between Supply Chains Partners. *European Journal of Operational Research*, 163, 668--687. (2005)
3. Grosz, B. N., Poon, T. C.: SweetDeal: Representing Agent Contracts with Exceptions Using Semantic Web Rules, Ontologies, and Process Descriptions. *International Journal of Electronic Commerce*, 8, 61--97(2004)
4. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5, 199--220 (1993)

5. Guarino, N.: Formal Ontology, Conceptual Analysis and Knowledge Representation. *International Journal of Human-Computer Studies*, 43, 625--640 (1995)
6. Guarino, N.: Understanding and building, using ontologies. *International Journal of Human-Computer Studies*, 46, 293--310. (1997)
7. Horrocks, I., Patel-Schneider, P. F., Harmelen, F. V.: From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Journal of Web Semantics*, 1, 7--26 (2003)
8. Horrocks, I., Patel-Schneider, P. F.: Reducing OWL entailment to description logic satisfiability. *Journal of Web Semantics*, 1, 345--357 (2004)
9. Horrocks, I., Patel-Schneider, P. F., Bechhofer, S., Tsarkov, D.: OWL rules: A proposal and prototype implementation. *Journal of Web Semantics*, 3, 23--40 (2005)
10. Kayed, A., Colomb, R. M.: Extracting ontological concepts for tendering conceptual structures. *Data & Knowledge Engineering*, 40, 71--89 (2002)
11. Lambert, D. M., Cooper, M. C.: Issues in Supply Chain Management. *Industrial Marketing Management*, 29, 65--83 (2000)
12. Lambert, D. M., Emmelhainz, M. A., Gardner, J. I.: Developing and Implementing Supply Chain Partnerships. *The International Journal of Logistics Management*, 7(2), 1--17 (1996)
13. Maloni, M. J., Benton, W. C.: Supply Chain Partnerships: Opportunities for operations research. *European Journal of Operational Research*, 101, 419--429 (1997)
14. Min, H., Zhou, G.: Supply Chain Modeling: Past, Present and Future, *Computer & Industrial Engineering*, 43, 231--249 (2002)
15. O'Connor, M. J., Knublauch, H., Tu, S. W., Grossof, B., Dean, M., Grosso, W. E., Musen, M. A.: Supporting Rule System Interoperability on the Semantic Web with SWRL. LNCS, vol. 3729, pp. 974--986. Springer, Heidelberg (2005)
16. Olhager, J., Selldin, E.: Supply Chain Management Survey of Swedish Manufacturing Firms. *International Journal of Production Economics*, 89, 353--361(2004)
17. Stadtler, H.: Supply Chain Management and Advanced Planning-basics, overview and challenges. *European Journal of Operational Research*, 163, 575--588 (2005)
18. Uschold, M., Grueninger, M.: Ontologies: principles, methods and applications. *The Knowledge Engineering Review* 11, 93--155 (1996)