

Building Dynamic Business Process in P2P Semantic Web

Timon C. Du

Department of Decision Sciences and Managerial Economics, The Chinese University of
Hong Kong. E-mail: timon@cuhk.edu.hk
and

Eldon Y. Li*

College of Commerce, National Chengchi University, Taipei 11605, Taiwan, and Orfalea
College of Business, California Polytechnic State University, San Luis Obispo, CA
93407, U.S.A. E-mail: eli@calpoly.edu

***Corresponding author**

Building Dynamic Business Process in P2P Semantic Web

Abstract:

Business process management systems such as the workflow management system and the enterprise application integration system manage process flow on a minute-by-minute basis in various application domains. In the conventional approach, the business process must be predefined before it is implemented. However, involving business users in the early stage of the design phase is neither efficient nor realistic in the dynamic business world. This study proposes a framework to implement a dynamic business process in the P2P Semantics Web, which provides the flexibility to dynamically alter business process and to take semantic data into consideration. The system is demonstrated by a case of a manufacturer that is processing an order.

Keywords: Business Process, workflow management system, P2P, Semantic Web

INTRODUCTION

In the past decade, the Internet technology boom has encouraged the development and sharing among people, organizations, and enterprises of vast ranges of information. However, most of that information is written in **Hypertext Markup Language** (HTML), which mainly follows a predefined format to express the content, where well-formatted information is normally written for human comprehension rather than machine automation. This means that when the information volume grows, the time to locate and digest the information increases even more rapidly. In this way, users need to make a tremendous effort to locate information that fits their needs. There are many possible solutions to resolve the information overloaded problem, such as paying money to the

search engine to improve visibility, which is called “**paid placement**” or “paid inclusion.” Sophisticated solutions such as allowing users to write a query paragraph rather than simply inputting keywords are also possible. In the years to come, we will see many more innovative solutions to the problem.

The Web has also evolved to become a service-providing medium. Web services use software applications to provide interoperability, whereby they discover, describe, and access other services from the Internet, **Intranets**, and **Extranets**. This leads to the adoption of **XML** (Extensible Markup Language) technology in which information is shared in text format. Note that XML provides independence of applications and data, which allows data to be shared among applications. However, the problem lies in determining what kinds of information can be shared and how it can be shared. As the Web is no longer a media for human-to-human communication because the information available is overwhelming, there is a need to seek help from machines in organizing and locating specific information. The evolution will take place in two dimensions: from syntactic to semantic and from static to dynamic. This will move the Web toward being a Semantic Web and Web service, and then advance the Web service into an intelligent Web service and the Semantic Web into Semantic Web services.

The Semantic Web structures Web content into semantic data for both humans and machines. The semantic data are the information and the meaning of the information. These are presented as structured collections of information and sets of rules. Although both the knowledge representation and rules have been studied for years in the area of artificial intelligence, the traditional approaches have rigid structures to ensure that new knowledge can be inferred from existing data and rules. In contrast, the decentralized

nature of the Semantic Web allows individual Web sites to represent knowledge in their own ways. This provides an opportunity for Web sites to grow independently and diversely.

This study will use the properties provided in the Semantic Web to build dynamic business processes, where “a business process is a collection of related structure activities that produce a specific outcome for a particular customer” (<http://en.wikipedia.org>). The process can be defined by attributes such as name, description, date, version, component, operation, etc. It is worth noting here the difference between workflow and business process. A **workflow** is a complex business process that normally involves many *tasks*; it is static and has to be well defined before applying. This also means that the *roles* assigned to specific tasks are predefined even though the *users* that are assigned to roles can be dynamically identified. A workflow can be activated many times, and each implementation is called a *case*, which has a unique identity and a limited lifetime. In this case, the business process is a concept mingling the workflow and the case because it indicates which task must be carried out for a specific customer. As workflow schemas are static and predefined, they are difficult to adopt in the rapidly changing environment, particularly for collaboration among partners (Zeng, et al., 2003). Moreover, problems of consistency after changing the workflow, resource optimization in the dynamic environment, the reuse of workflow, the workflow engine that is used to manage the changing workflow, and the flexibility to adapt workflow to new environments are all our concerns (Cichocki, and Rusinkiewicz, 2004).

In this study, the business process is developed in the **peer to peer** (P2P) architecture of the Semantic Web. The **P2P architecture** is used as a communication

platform in which each party (**node**) has a similar role and a similar ability to handle business processes with other parties directly or through a community service (<http://searchnetworking.techtarget.com>). Moreover, each node is built with Semantic Web technology that allows the semantics of Web content to be included in the handling of business processes. The advantages of this approach are: (1) Web sites that participate in the business process are dynamically selected (for example, the selection of suppliers); (2) information and support for implemented tasks are dynamically collected (for example, order processing); (3) the implementation of business processes can be automated (such as in outsourcing); and (4) the business process can be dynamically modified (such as in business process collaboration). Figure 1 exhibits an example of how a business process is dynamically modified. The remaining sections are organized as follows. Section 2 briefly introduces the links between business and the Semantic Web. Section 3 illustrates the architecture. Section 4 presents a demonstration, and Section 5 contains the discussion.

***** please put Figure 1 about here *****

Business Process and Semantic Web

A business process defines the activities, including input, output, and method, that are necessary to complete a specific assignment. A business process is created to deal with a specific *case* in an organization, such as applying for a mortgage, placing orders or proceeding with engineering tests. The structure of a business process can be very complex. It can be composed of many sub-processes (also called processes), such as a

workflow. The work in a business process can be identified as a *task*, which represents the indivisible unit of work. The task is carried out by a *process*. When the processes are carried out in a business process they follow a specific sequence that determines which task needs to be performed next. A case should be completed within a certain time limit and should exit the system when the work is completed. This also means that attributes are needed to describe the state and content of the process.

There are four different types of sequences in a complex **business process**: *sequential*, *parallel*, *selective*, and *iterative* routings. Sequential routing confines one task to be executed before another task, whereas parallel routing allows two tasks to be performed without feedback from each other. Selective routing provides a choice between or among tasks, and iteration allows the same task to be performed more than once.

During implementation, the process must be enacted to perform a task. Tasks are assigned to designated *roles* following principles such as the separation of duties, least privilege assignment and data abstraction (Sandhu, et al., 1996). These principles ensure the successful implementation of the business process. For example, the separation of duties assigns two sensitive tasks to two exclusive roles so that conspired perpetration can be avoided. This principle is easily adopted within an organization (by using an Intranet) and can be accepted in supply chain collaboration (by using an Extranet), but it is difficult to use on the Internet, in which the new assignments can go to any participant. In a **client-server** type of community, the duty to maintain the principle is normally assigned to a community server, which becomes a bottleneck because it needs to trace every business process. This study uses P2P architecture to develop the business process.

As stated earlier, P2P architecture offers the advantage of sharing workloads (King, Ng, and Sia, 2004). Hence, the duty to maintain the principle should fall on any node in the community that uses the business process. Similarly, the least privilege policy, also called the ‘need to know’ policy (Castano, et al., 1995), provides only minimum information for completing the task. Information sharing between nodes is also based on this principle, whereby only the minimum information is carried via agents that roam between nodes. However, as information is limited, when one node receives separate information from two sources that belong to the same business process that information cannot be composed back to the original message, which means that sensitive data can be leaked.

A **business process management system** manages the process flow on a minute to minute basis in various application domains such as office automation, finance, healthcare, telecommunication, manufacturing and production (Bertino, Jajodia, and Smarati, 1999). As has been discussed, in conventional business process systems, such as the **workflow management system** (WFMS), the process is static and predefined. However, it is unrealistic to apply a static and predefined process in a dynamic business world. Much research has tried to resolve dynamic business process problems. Examples of these problems include (1) evolving the process when the system parameters, the relationships among parameters, or communication among different partners in the supply chain changes (Rouibah and Caskey, 2003); (2) differentiating a shared public workflow from a private workflow to alleviate the effect of a change in a public workflow on a private domain (van der Aalst, 2003); (3) using different agents to manage the variation of workflows, in which the inter-agent workflow concerns control flow, data flow, and material flow between agents, and the intra-agent workflow provides flexibility

in performing the activities on the work-list (Zhuge, 2003); (4) applying knowledge-based techniques to make workflow systems more adaptive and to provide more flexible process management (Chung, et al, 2003); and (5) not forming the business process until runtime to tolerate the dynamics (Zeng, et al., 2003; Su, et al., 2003)

The **Semantic Web** brings structure to the meaningful content of Web pages, whereby software agents roam from one page to another to carry out sophisticated tasks for users (Berners-Lee, Hendler, and Lassila, 2001) . The implementation of the Semantic Web should be decentralized, with structured information and tasks carried out by the software agents, and with sets of inference rules helping the software agents to retrieve information. Most current studies on the Semantic Web focus mainly on the services carried out by software agents, such as ontology design, the visual environment for browsing, **RDF** (Resource Description Framework) **Model and Syntax Specification**, and Web Services with ontologies, as opposed to individual servers (<http://www.w3.org/>). Less attention has been paid to the coordinated efforts of the servers to form a group ideology.

A Semantic Web is similar to a deductive database and a knowledge-based system, where new knowledge is deduced by applying rules to existing facts (Du, 2002). The differences between these systems are that the knowledge-based system does not use secondary storage to house data as do deductive databases and the Semantic Web, and both the knowledge-based system and deductive database apply rigid structures to the knowledge representation and deduction rules to ensure that new knowledge can be obtained from inference. In contrast, the Semantic Web allows websites to represent the knowledge and the relationships of the knowledge in their own ways. The information on

websites can then be shared for use in different domains. For example, a query can be interpreted by the semantics interpreter through the query statement in such a way that the query results are more relevant (Jain, Aparicio, and Singh, 1999; Karvounarakis, et al., 2003).

In the Semantic Web, the **software agent** is used to roam from page to page to carry out tasks that are assigned by users. However, it can do more than being a messenger because a mobile agent is an autonomous object that is created for dynamic and distributed applications that execute designated tasks. Referring to Wooldridge (Wooldridge and Jennings, 1995), agents can be identified as either *strong* or *weak*.

Strong agents are capable of mentalist notions, rationality, veracity, adaptability, and learning. These capabilities come mainly from the technology of artificial intelligence.

Weak agents, in contrast, can complete tasks autonomously, interact with external objects, and are reactive or proactive toward environmental change based on a pre-planned scheme. Software agents have been implemented in many distributed environments to share system loading and increase flexibility. Several applications have been successfully developed using mobile agents: for example, the supply chain SMART project (<http://smart.npo.org>), the virtual enterprise (Jain, Aparicio, and Singh, 1999), information retrieval (Cabri, Leonardi, and Zambonelli, 2000), the Internet-based auction house (Sandholm, and Huai, 2000), secured transactions (Castano, et al., 1995), and distributed network management (Du, Li, and Chang, 2003). Some studies have further integrated mobile agents with CORBA, such as MESIS resource management (Bellavista, Corradi, and Stefanelli, 2000) and **broadband intelligent networks** (Chatzipapadopoulos, Perdikeas, and Venieris, 2000). In general, the software agent

system can be applied to the areas of electronic commerce, personal assistance, secure brokering, distributed information retrieval, telecommunication network services, workflow applications and groupware, monitoring and notification, information dissemination, and parallel processing (Lange and Oshima, 1998).

The Architecture of Semantic Business Process

To implement a P2P framework, a node must first download a P2P networking program, such as IBM's **Advanced Peer-to-Peer Networking** (APPN), and a semantic business process program, such as **Semantics Flow Builder** (SFB) and mobile agents. The SFB supports four functions – **Implementation Manager**, **Process Planner**, **Resource Manager**, and **Process Monitor** – and three mobile agents – a **configure agent**, a **runtime agent**, and a **supervise agent** – as shown in Figure 2. The three agents and four SFB functions interact as shown in Table 1.

***** please put Figure 2 and Table 1 about here *****

SFB functions

Four SFB functions are responsible for different duties. The *Process Planner* is used to design business processes and assign the processes to roles (Web sites). The Planner sends the configure agent to search for another network member that can implement the next task through community directory services (similar to UDDI). Once the agent finds another qualified network member, it connects to that member and then carries the job assignment to the website (on a need-to-know basis). If none of members (Web sites) qualifies, then the Planner aborts the process. If more than one candidate Web site is

found, then the configure agent activates the Resource Manager to determine which website should be chosen (the separation of duty is sustained at this moment).

When a new task arrives at the next member, the Process Planner of the new Web site activates a runtime agent, which is dispatched to collect data, report to the initiated Web site whenever needed, and execute the task and work locally with the Implementation Manager (which will be explained later). When the task is completed, the member either returns the work to the initiated website or searches for a new member to continue the work, depending on the business process. In the latter case, a new configure agent is sent to the community directory service to search for another member that can be the candidate for the next task. This procedure repeats until it reaches to the end of the process blueprint designed by the initiated Web site. Meanwhile, a member may need to interact with previous members or the initiated Web site if it needs more information to execute its work. When a Web site encounters the completion of the process, it should report to the website that activated it (stacking).

It should be noted that this design allows a member to dynamically modify the business process according to local needs. This is based on the understanding that the local member should have a better knowledge of how to implement a particular task than any other member. However, when authority is given to a Web site, the business process can grow unpredictably. Therefore, it is the duty of the local Process Planner to check the soundness if the website extends the business process (this will be discussed later.) Fortunately, based on the hierarchical conceptualization of workflow theory, a complex workflow can be divided into many sub-flows, and if the sub-flows are sound the parent flow is also sound (Aslst and Hee, 2002). Formally, business processes are modified as:

Business process $BP(W, T, F)$ where W is the finite set of websites, w ; T is a finite set of tasks, t , ($W \cap T = \emptyset$); and $F \subseteq (W \times T) \cup (T \times W)$ is a set of flow relations.

If $inf(M)$ is a strictly increasing function of the distance between current belief τ' and original belief τ ; $inf(M) = |\tau' - \tau| = \sum_{i=1}^m (\tau_i - \tau'_i)^2$ is the change of belief due to semantic information; and $t_i \notin t$ where t_i is the new task to implement τ'

then $BP \subset BP'$ where $F' \subseteq (W' \times T') \cup (T' \times W')$ and $w \subseteq w'$

dispatch Configure Agent

activate Resource Manager

call Reachability Graph checking

endif

The *Resource Manager* is responsible for assigning activities to resources. This manager should interact with the Implementation Manager to execute the task and the Process Monitor to ensure that constraints are upheld. Moreover, the Resource Manager has guidelines (such as separation of duty) to determine the allocation of sources when more than one Web site is found. This is where the Semantic Web can play a significant role. In general, the guidelines can be as simple as providing rules of thumb, such as the cheaper the better or the faster the better, or more advanced rules, such as providing resource clustering, pre-conditioning, exclusion, or case attributes. The selection should refer to the semantic data. Note that the semantics refers to the meanings and the logic of the requisition. It is important for the Resource Manager to make the right selection based on sufficient information.

The *Process Monitor* oversees the activities of the business process and uses the Supervise Agent to communicate with participating members. The main duties include

alerting the Web site if the task is delayed and aborting the process if no member can be assigned to the next task.

The duty of the *Implementation Manager* is relatively straightforward: it completes the task assignment. The task can be executed immediately, at a preset time, or after a series of events. When the assignment is completed, the Implementation Manager informs the Process Planner to dispatch a Configure Agent to locate Web site for the next task assignment. However, the Implementation Manager notices the encountered circumstance and needs to consider more factors: i.e. if the information $inf(M)$ is a strictly increasing function of the distance between current belief τ' and original belief τ , then the Implementation Manager activates the Process Manager again, and the business process can be modified accordingly.

Mobile agent function

Mobile agents travel around websites as messengers or information locators. In the design, a *Configure Agent* locates a candidate website for the next process and providing instructions to the new selected member. That is, the Configure Agent takes the blueprint of a business process, the participant list, the data for implementing the task, and the duty and instructions of task assignment to the next Web site. When it arrives at a new Web site, the local Implementation Manager is enacted immediately, at a preset time or when certain events occur.

The second agent is a *Runtime Agent*, which is responsible for supporting the execution of a business process. It is the duty of the Runtime Agent to obtain augmented information to support the execution of the Implementation Manager. In a P2P Semantic Business Process, every participating member relies on the monitoring function.

As the participating members are dynamically selected and the process flow is modifiable, a *Supervise Agent* is used to monitor the activities of the members based on the most current status reported to and stored in the Process Monitor of the initiated Web site. That is, the Supervise Agent is sent by the initiated Web site to the member Web sites periodically (predefined by timers) or actively (triggered by events) to ensure that the business process operates smoothly.

The advantages of a semantic business process

The semantic business process takes advantage of the Semantic Web to implement business processes. The first advantage of this approach is a flexible data structure. The conventional Web page presents content and links to a database when a large volume of information is associated with it. As the databases are all using the relational data model, the data structure is rigid and it is difficult to provide personalized service. However, this is not the case in Semantic Web, where data is stored in the format of XML/RDF and is suitable for dynamic environments.

The second advantage is that the Semantic Web can act as a medium to support decision-making rather than simply to provide information, as does the conventional **HTML** Web page. This is because the content that is embedded in the RDF provides additional information in its tag that can be directly employed in user applications.

The third advantage is that the conventional Web allows users to specify keywords for searching relevant content. However, in the Semantic Web, the system knows more about both information providers and requesters, and then matches the information that is most useful for the requesters. For example, when a user searches for “industrial

engineering” on the Web, university departments such as “system engineering” may not appear in the results, even though the courses offered are actually close to what the user is looking for. In the Semantic Web, the search can be further conducted against the course descriptions and program tags, and “system engineering” results can also be included.

IMPLEMENTATION AND DEMONSTRATION

This section demonstrates how the business process can be built and expanded in the Semantics Web. The example shown in Figure 3 represents a PC manufacturer receiving orders from buyers. The manufacturer first checks its inventory for finished products, work in process, and the capacity of its own factories. If it cannot fill the orders from its own capacity, then the manufacture outsources all or a part of them to collaborative partners. Depending on the semantic data, such as quality and collaboration history, the system recommends different supplier lists to the manufacturer.

***** please put Figure 3 about here *****

Web Ontology and Semantics

The ontology defines the meaning and the relationship of a particular subject area. To implement the Semantic Web, the **ontology** of the subject area must be pre-defined.

The **Resource Description Framework** (RDF) can be used to represent machine-processable information in the World Wide Web using a triple format: subject,

predicate, and object (<http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>). RDF uses XML to allow users to define their own document formats to represent statements.

In this demonstration, RDF mainly defines database content that stores information about products, suppliers and the community. The content in these files can be embedded into Web pages to create “semantic” Web files. Here, we separate the content into four RDF files for demonstration. All the RDF files have been validated by the W3C RDF validation service at <http://www.w3.org/RDF/Validator/>. The structure graph of the major data fraction representation is shown in Figure 4.

- (1) Storage.rdf stores data about the finished products and work-in-process products that are either current inventory items or available-to-promise items.
- (2) DeptN.rdf stores data about manufacturing departments and production capacities in recent weeks.
- (3) SupplierN.rdf stores data about collaborative partners, such as locations, prices, specifications, and stocks of different products.
- (4) Resource-manager.rdf acts as a coordinator to determine the selection of collaborative partners. The file also contains information that is related to the collaboration history and ratings of the suppliers.

***** please put Figure 4 about here *****

System Development

This system was developed with the following tools: (1) Macromedia **Dreamweaver** MX for building common HTML files and dynamic **JavaServer Pages** (JSP) Web pages; (2)

Jbuilder X for building Java files and classes (JSDK 1.4.3 included); (3) Jena 2.1 for building Semantic Web applications, including creating, modifying and querying RDF files, and providing a programmatic environment for RDF, RDFS, and OWL, including a rule-based inference engine; (4) RDQL for querying RDF content; and (5) Tomcat 5.1 for building JSP files to execute on a server.

Four classes with many methods have been built, and the use case is shown in Figures 5 and 6.

(1) Class Storage is designed to store Information about the product inventory. Two methods are used:

- `getStock(String product)`: to get the number of finished goods of the specified product in stock; and
- `getWip(String product)`: to get the number of works in process in stock of the specified product.

(2) Class Department is designed to store information about manufacturing plants. Two methods are used:

- `getCapacity(String week)`: to get the capacity of the specified week; and
- `getCapacity(int n)`: to get the total capacity of n weeks from now.

(3) Class Resource Manager is designed to store Information about the collaborative partners. One method is used:

- `getHighScore(String product)`: to find the supplier with the highest score under the current conditions.

(4) Class Supplier is designed to store information about selected suppliers. Four methods are used:

- getPrice(String product): to get the price offered by the supplier of the specified product;
- getStock(String product): to get the stock of the specified product that the supplier owns;
- getRank(): to get the ranking of the supplier, which is stored on the community page (and thus suppliers are not allowed to modify their own scores); and
- calScore(String product): to calculate the score for suppliers under different conditions.

***** please put Figures 5 and 6 about here *****

Illustration

This section uses the example in Figure 3 for illustration. The scenario is that a manufacturer checks the inventory and work-in-process products to ascertain whether the available-to-promise amount is sufficient when a new order is placed. The manufacturer has three plants, each has a different capacity and inventory. If the internal capacity cannot satisfy the order, the purchasing department of the company will outsource a part or all of the order to collaborative partners. Assuming that the current inventory of available-to-promise finished product is 100 and of work-in-process product is 100, then the manufacturing lead time is one week. In Figure 3, node A represents the Implementation Manager of the manufacturer, node B represents the Resource Manager of the manufacturer, and node C represents the Implementation Manager of suppliers.

Scenario 1: The order amount is 50 and the delivery date is two weeks from now.

After receiving an order, the company first checks whether or not the inventory of finished goods in stock is sufficient. If the finished goods in stock are enough to cover the order, then it will be confirmed. In this scenario, the finished product is sufficient to promise the order, and the order is thus confirmed.

Scenario 2: The order amount is 150 and the delivery date is two weeks from now. As the finished product is insufficient, the system will check the work-in-process product. In this scenario, adding the amount of the finished product and work-in-process product is sufficient to promise the order. The order is confirmed.

Scenario 3: The order amount is 900 and the delivery date is two weeks from now. As the available-to-promise amount is not enough, the system will check whether the 3 production plants (plants 1-3) are able to produce the outstanding amount within the required delivery time. If they can, then jobs will be assigned to the plants according to their remaining capacities in the required time. That is, the order will first go to the plant with the largest remaining capacity. For example, as shown in Table 2, in the two week period, plant 1 can produce 300, plant 2 can produce 300, and plant 3 can produce 500. As the capacity is sufficient to meet the order, the order is confirmed. The shortage of this order is 700 (order – inventory – work in process), and the order will be split into 500 units for plant 3 and 200 units for plant 1.

***** please put Table 2 about here *****

Scenario 4: The order amount is 1400 and the delivery date is two weeks from now. The demand is larger than the remaining capacities and the inventory of the three plants,

so the purchasing department searches for suppliers to outsource the order. The system sends Configure Agents to the community directory to find a list of suppliers that have sufficient stock to cover the outstanding amount (order amount – finished goods – work in process – maximum amount that the departments can produce). As more than one supplier can supply the products, the system will recommend a list of best matching suppliers based on the “semantic” information written in RDF/XML files. For example, there are many factors, such as price, reputation, collaborative history, and current stock, to consider when choosing a supplier. After determining the supplier, the Process Planner of the purchasing department will alter the current business process to include other business processes, such as quality control and delivery service, from the supplier. Again, it is the duty of the purchasing department to check the soundness of the business process. The order is then confirmed.

Table 3 shows the data that is used in this demonstration. When more than one supplier is qualified, it is the task of Resource Manager to prioritize the suppliers. The algorithm is as follows: if the outstanding amount is less than 10% of the total order, then price will be considered as the most critical factor and given a weight of 50%, with stock given a weight of 30% and reputation given a weight of 20% for selecting the supplier. If the outstanding amount is more than 10% but less than 20% of the total order, then stock is considered to be the most critical factor (50%), followed by price (30%) and reputation (20%). If the outstanding amount is more than 20% of the total order, then reputation becomes the most critical factor (50%), followed by stock (30%) and price (20%). The rationale is that when the outsource amount is very large, the manufacturer pays special

attention to elements of reputation such as technology level, quality, and experience. The calculation is conducted by assuming that three suppliers all meet the requirements.

***** Please put Table 3 about here *****

The price, stock and reputation of the three suppliers are represented as p_1 , p_2 , and p_3 , s_1 , s_2 , and s_3 , and g_1 , g_2 , and g_3 . The higher score is given to the higher preference, such as low price, high stock, and good reputation. The total score is then multiplied by the weights. In this scenario, as the 1300 units can be produced by the 3 plants in two weeks, 100 units will be outsourced. Therefore, price is considered to be the most critical factor as the outstanding amount is 7% of the total order. The system thus places supplier 001 at the top of the recommendation list. However, the list is only given to the Resource Manager of the manufacturer as a reference. The Resource Manager should refer to other criteria such as the collaboration history of the suppliers.

Scenario 5: The order amount is 1500 and the delivery date is two weeks from now. As the outsourced amount is 200 units (13% of the total order), current stock is considered to be the most critical factor. The system thus recommends supplier 002 first.

Scenario 6: The order amount is 1700 and the delivery date is two weeks from now. As the outsourced amount is 400 units (23% of the total order), reputation is considered to be the most critical factor. The system thus recommends supplier 001 first.

CONCLUSIONS AND FUTURE STUDY

This study has argued that current business process systems, such as the workflow management system, need to define the whole process before implementing it. This, however, is not suitable for the dynamic business world because collaboration is dynamic. This study has presented a system that uses the Semantic Web to build dynamic business processes. The system is built on top of a P2P architecture that allows each Web site (member) to operate independently and collaboratively through the community directory. In this design, each member has the functions of Process Planner, Resource Manager, Process Monitor, and Implementation Manager. Three mobile agents are used for the communication: a Configure Agent, a Runtime Agent, and a Supervise Agent. The system is illustrated with the case of the order handling process of a manufacturer. The illustration demonstrates that the business process can be extended when necessary, and that the operation of the business processes can take semantics data, defined in RDF and XML, into consideration.

This study does not consider areas of security such as access control. Future studies could address the issues of sharing information when the business process is extended indefinitely. Moreover, the directory service is relatively simple in this study. A related future study on Web service could be used to improve this issue. Another plausible issue is how to select suppliers when thousands of websites are qualified or when business processes are interrelated.

References

Aslst, W. and Hee, K. (2002). *Workflow Management: Models, Methods Systems*,
Cambridge, MA: MIT Press.

- Bellavista, P., Corradi, A. and Stefanelli, C. (2000). An Integrated Management Environment for Network Resources and Services, *IEEE Journal on Selected Areas in Communications* 18, no. 5, 676-685.
- Berners-Lee, Tim, Hendler, James, and Lassila, Ora. (May,2001). The Semantic Web, *Scientific American*, 284, no. 5, 34-44.
- Bertino, E., Jajodia, S. and Samarati, P. (1999). A Flexible Authorization Mechanism for Relational Data Management Systems. *ACM Transactions on Information Systems* 17, no.2, 101-140.
- Cabri, G., Leonardi, L. and Zambonelli, F. (February 2000). Mobile-Agent Coordination Models for Internet Applications, *IEEE Computer* 33, no. 2, 82-89.
- Castano, S., Fugini, M., Martella, G. and Samarati, P. (1995). *Database Security*. Harlow, England UK: Addison-Wesley and ACM Press.
- Chatzipapadopoulos, F., Perdikeas, M. and Venieris, L. (June 2000). Mobile Agent and CORBA Technologies in the Broadband Intelligent Network, *IEEE Communication Magazine* 38, no. 6, 116-124.
- Chung P.W.H., Cheung L., Stader J., Jarvis P., Moore J., Macintosh A. (April 2003). Knowledge-based process management-an approach to handling adaptive workflow, *Knowledge-Based Systems* 16, no. 3, pp. 149-160.
- Cichocki, A. and Rusinkiewicz, M. (2004). Providing Transactional Properties for Migrating Workflows, *Mobile Networks and Applications* 9, 473-480.
- Du, Timon (June 2002,). Techniques and Applications of Emerging Database System Architectures, *Database and Data Communication Network Systems*, Edited by: Cornelius T. Leondes, Academic Press.

- Du, Timon, Li, Eldon, and Chang, An-Pin (July 2003). Mobile Agents in Distributed Network Management, *Communications of ACM*, 46, no. 7, 127-32.
- Goasdoué François and Marie-Christine Rousset (August 2004). Answering queries using views: A KRDB perspective for the semantic Web, *ACM Transactions on Internet Technology (TOIT)* 4, no. 3, 255-288.
- Jain, A., Aparicio, M. and Singh, M. (March 1999). Agents for Process Coherence in Virtual Enterprises, *Communications of the ACM* 42, no. 3, 62-69.
- King, I., Ng, C. H. and Sia, K. C. (July 2004). Distributed Content-Based Visual Information Retrieval System on Peer-to-Peer Networks, *ACM Transactions on Information Systems* 22, no. 3, 477–501.
- Karvounarakis, G., Magganaraki, A., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M., and Tolle, K., (August 2003). Querying the Semantic Web with RQL, *Computer Networks* 5, 42, no. 5, 617-640.
- Lange, D. and Oshima, M. (1998). *Programming and Deploying Java Mobile Agents with Aglets*, Reading, Massachusetts Addison-Wesley.
- Rouibah K. and Caskey K. (September 2003). A workflow system for the management of inter-company collaborative engineering processes, *Journal of Engineering Design* 14, no. 3, 273-293.
- Sandholm, T. and Huai, Q. (March/April 2000). Nomad: Mobile Agent System for An Internet-Based Auction House, *IEEE Internet Computing* 4, no. 2, 80-86.
- Sandhu, R. S., Coyne, E. J., Feinstein, H. L. and Youman, C. E. (February 1996). Role-Based Access Control Models, *IEEE Computer*, 38-47.
- Su, S.Y.W., Meng, J., Krithivasan, R., Degwekar, S. Helal, S. (January 2003). Dynamic

- Inter-Enterprise Workflow Management in a Constraint-Based E-Service Infrastructure, *Electronic Commerce Research* 3, no. 1-2, 9-24.
- van der Aalst W.M.P. (October 2003). Inheritance of Interorganizational Workflows: How to Agree to Disagree without Loosing Control? *Information Technology and Management* 4, no. 4, 345-389.
- Wooldridge, M.J. and Jennings, N.R. (February 1995). Agent Theories, Architectures and Languages: A Survey, *Lecture Notes in Computer Science* 890, 1-39.
- Zeng L., Benatallah B., Lei H., Ngu A., Flaxer D. and Chang H. (June 2003). Flexible Composition of Enterprise Web Services, *Electronic Markets* 13, no. 2, 141-152.
- Zhuge H. (May 2003). Workflow- and agent-based cognitive flow management for distributed team Cooperation, *Information and Management* 40, no. 5, 419-429.

Table 1 The Correlation of Builder Functions and Mobile Agents

	Process Planner	Resource Manager	Process Monitor	Implementation Manager
Configure Agent	✓	✓		✓
Runtime Agent	✓			✓
Supervise Agent			✓	

Table 2 The Capacity and Work-in-Process of Three Plants.

Plant	Capacity	Week 1	Week 2	Week 3	Week 4	Week 5
1	1000	900	800	700	500	600
2	800	600	700	600	500	400
3	800	500	600	600	400	500

Table 3 The Price, Stock, and Reputation of Three Suppliers.

Supplier		Price	Stock	Reputation
1	Product001	10	100	5
	Product002	20	150	
1	Product001	15	200	4
	Product002	25	200	
3	Product001	20	150	4
	Product002	20	200	

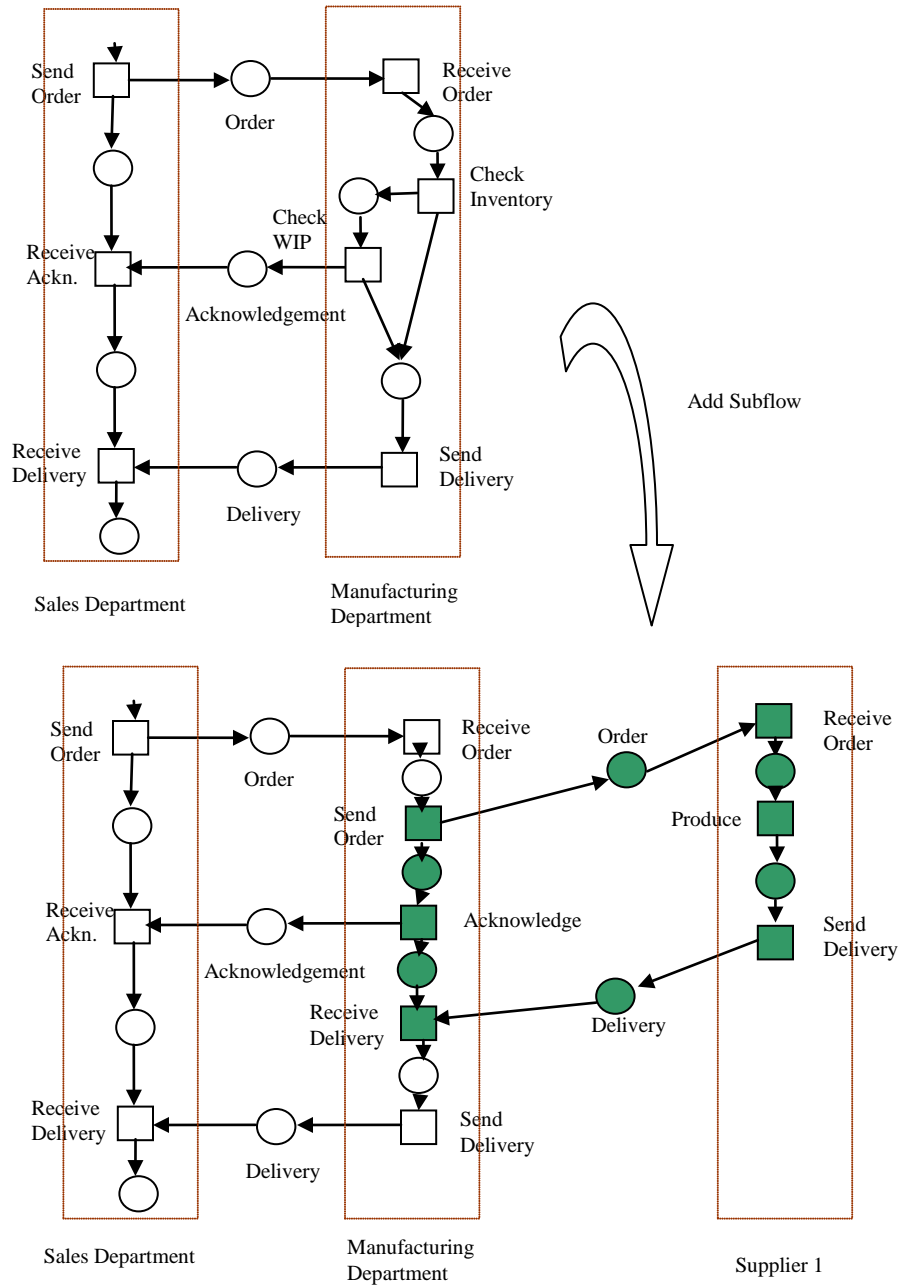


Figure 1 An Example of Dynamically Modifying a Business Process.

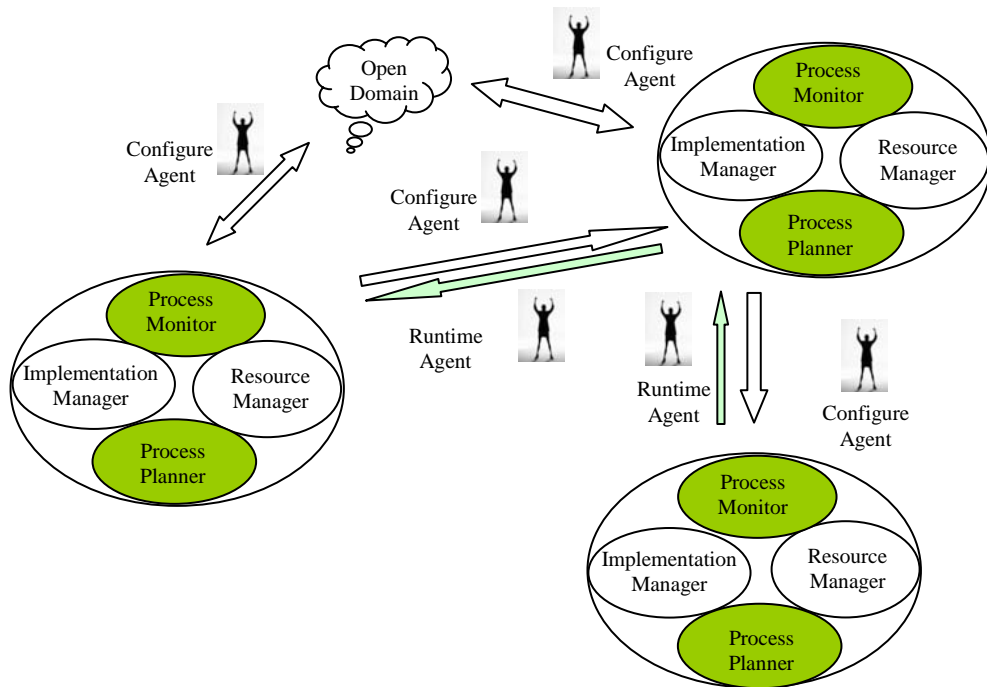


Figure 2 The Architecture of the Semantics Business Process

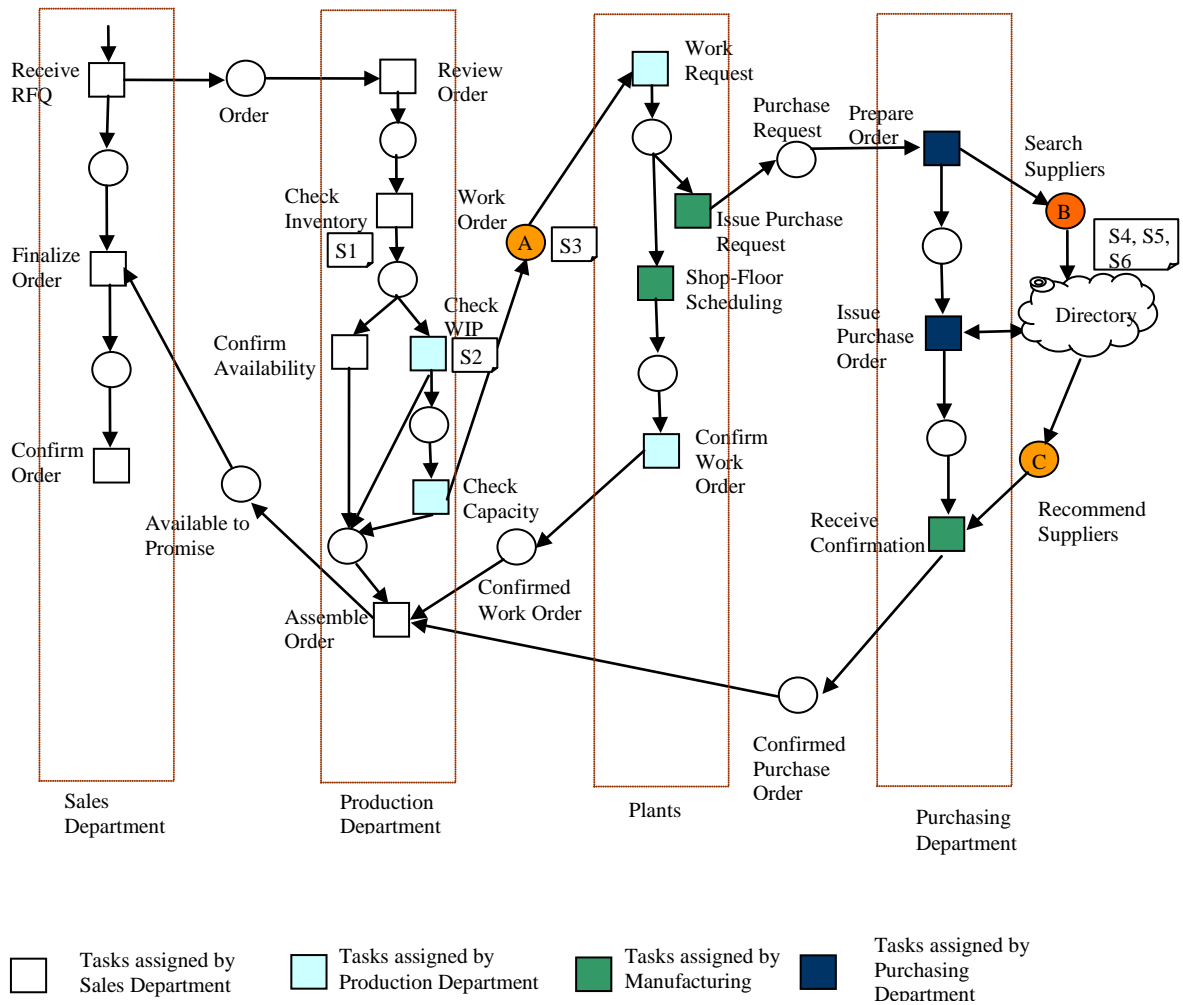
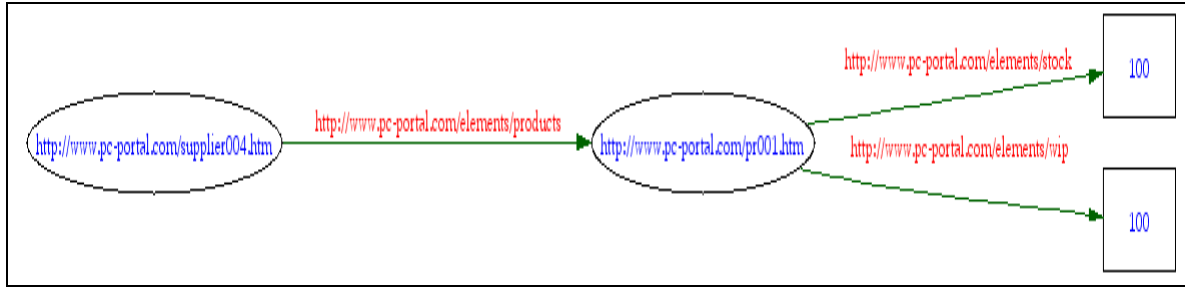
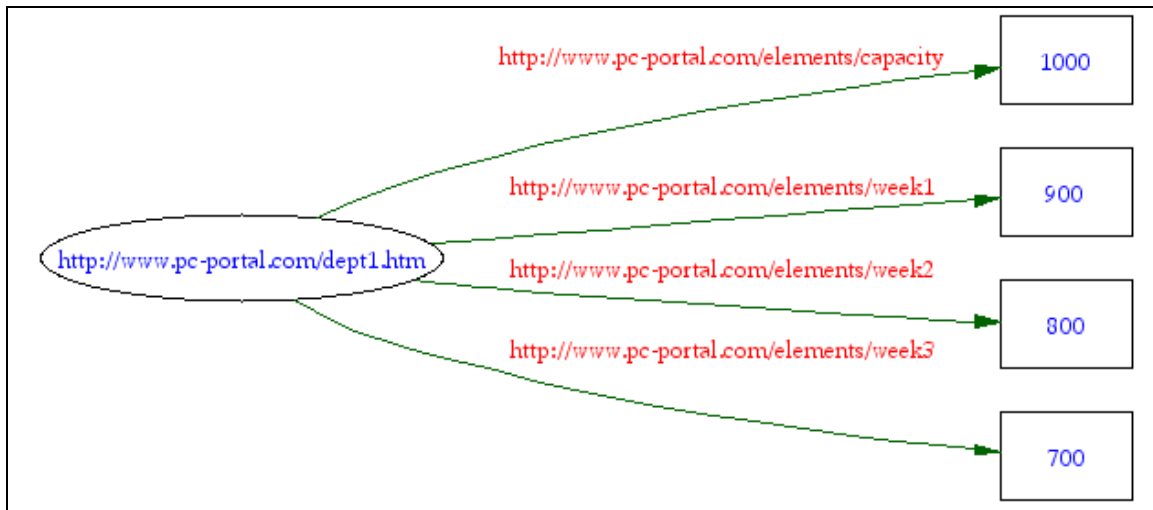


Figure 3 An Illustration of the Order Fulfillment of one Manufacturer.

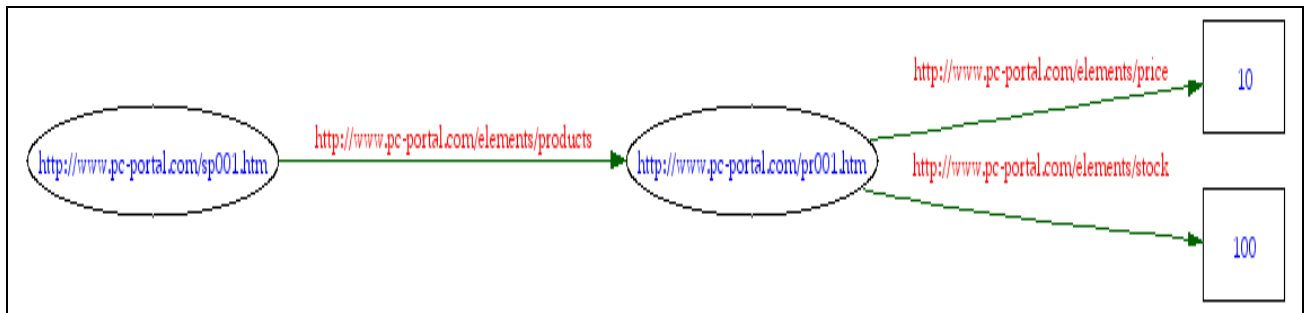
Storage: storage.rdf



Manufacturing plant: deptN.rdf



Supplier: supplierN.rdf



Resource Manager: resource-manager.rdf

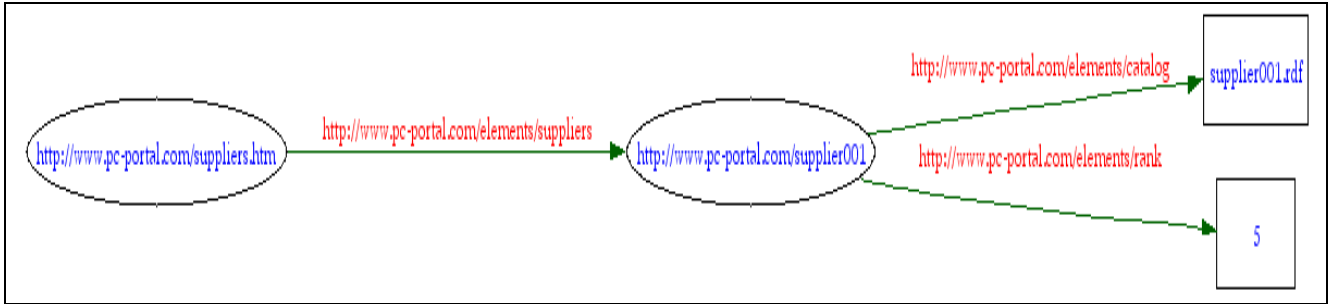


Figure 4 Graphical Presentation of four RDF Files.

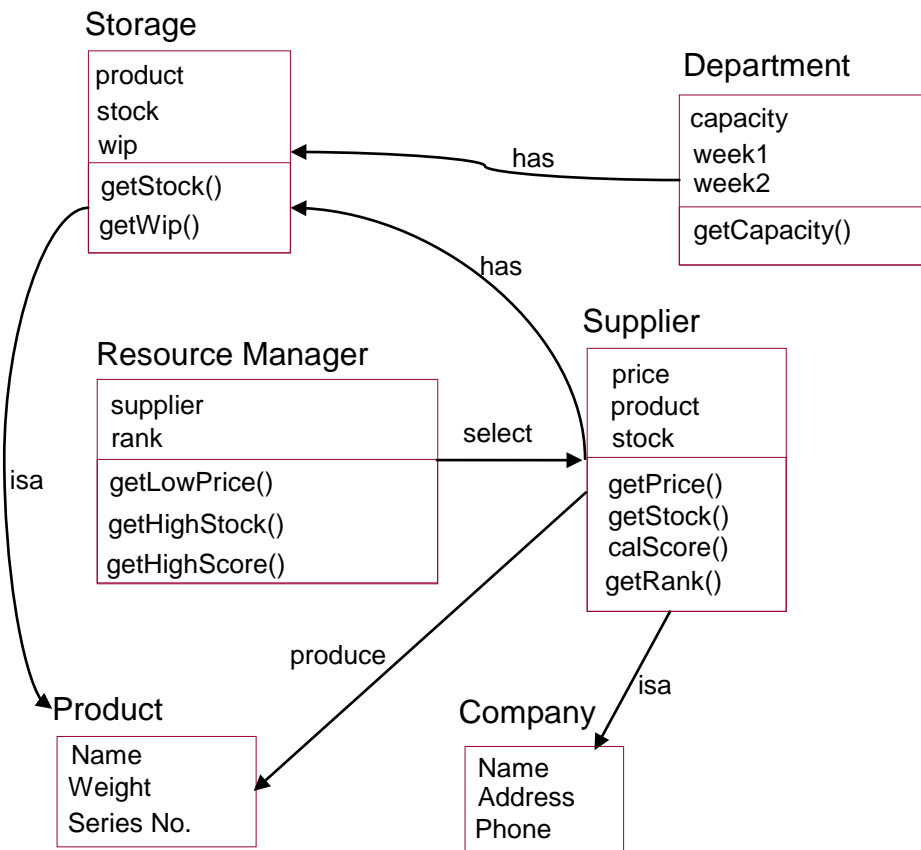


Figure 5 Graphical Ontology of the Order Placing Process.

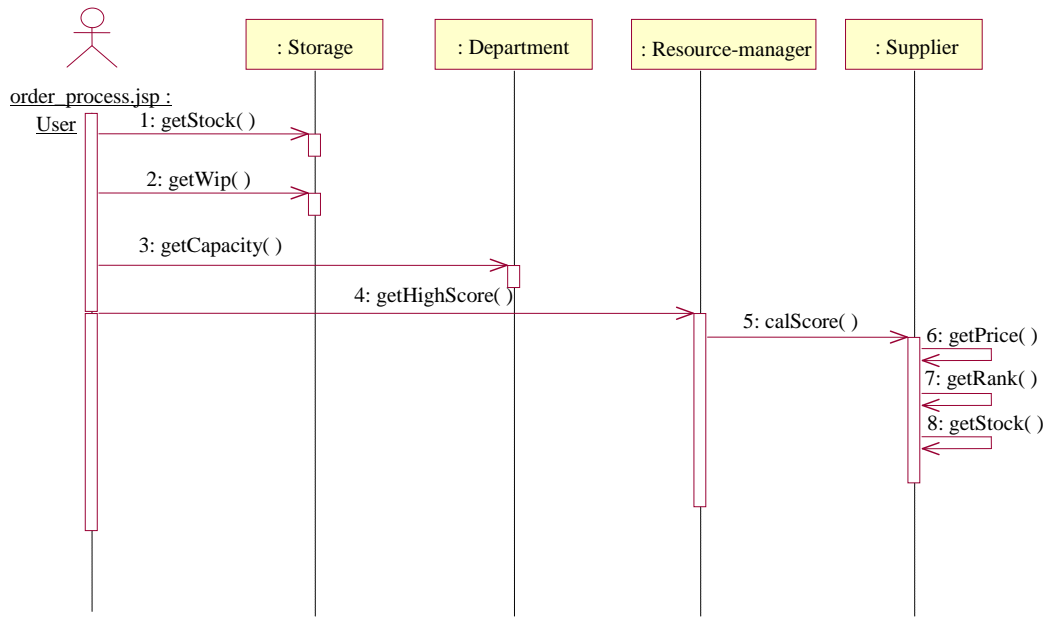


Figure 6 A Graphical Use Case Presentation

