On Demand Integration of Dynamic Supply Chain Application Based on Semantic Service Oriented Architecture

Juanqiong Gou[1], Xi Yang[2] and Wei Dai[3]

[1, 2] School of Economics and Management, Beijing Jiaotong University, Beijing 100044, P.R. China jq_gou@263.net 06120693@bjtu.edu.cn
[3] School of Information Systems, Victoria University, Australia Wei.Dai@vu.edu.au

Abstract. Dynamic Supply Chain is flexible to support various needs of enterprises. The integration and collaboration of enterprise systems offer significant challenges, while the problem is the difficulty for the integration of distributed heterogeneous information system as the partnership changes. The emerging technologies of Service-Oriented Architecture (SOA) and Semantic Web offer promising solutions by enhancing the reusability and interoperability of different systems and promoting cooperation among business partners.

This paper analyzes the requirements for On-Demand Integration of supply chain applications and discusses various integration solutions and their challenges. The analysis indicates that an approach combining the Semantic Grid and SOA technologies will be an appropriate solution. Therefore, a generic conceptual architecture for the integration by using Semantic Grid enabled SOA approach is proposed. An example scenario is presented to show a potential application of this architecture with a prototype adopting the WSMX (Web Service Modeling Execution Environment) as the basis of its implementation and deployment.

Keywords: *Dynamic supply chain, On-Demand integration, Semantic grid, Service-oriented architecture, Ontology, WSMX (Web Service Modeling Execution Environment)*

1. INTRODUCTION

The rapid changes of external environments and the uncertain factors within an enterprise (such as malfunction of manufacture equipments, delayed delivery of raw materials, etc.) put significant pressure on the company's performance, and at the same time offering collaboration opportunities among business partners along the supply chain. The core business partners can be formed dynamically according to customers' orders.

Dynamic Supply Chain (DSC) usually contains different cooperation relationships covering from upstream raw material suppliers, manufactures that play a core role in the process, to downstream distributors and retailers. Although the business agreements are reached prior to any business deals among the business partners in a B2B scenario, multiple delivery channels may be available for a buyer business. Therefore, on-demand supply chain channels are generated (dynamically) to meet individual business needs. Figure 1 describes the company relationships along the supply chain which consists of several supply chains distinguished by distinct colors initiated by OrderA , OrderB and OrderC respectively.

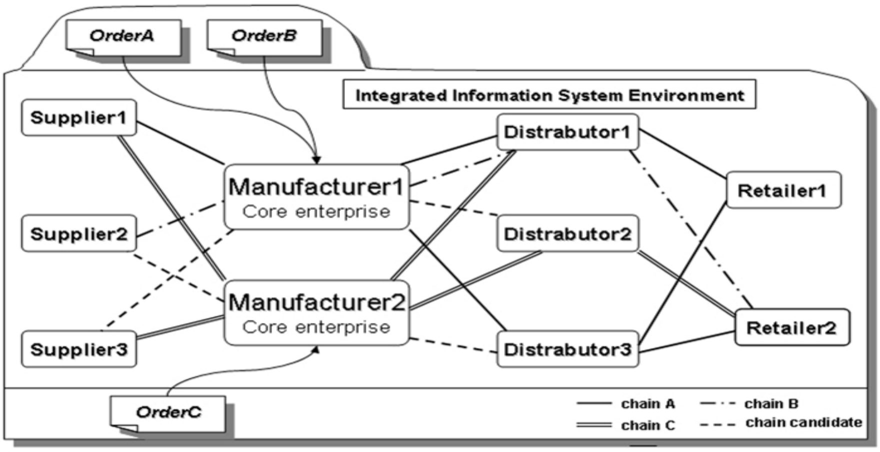

Figure 1. **Dynamic Supply Chain Practice**

To achieve the automated formation of a dynamic supply chain and enable the enterprises to cooperate effectively, there still exist two cruxes of the matter awaiting settled.

Firstly while manufacturers searching for the potential partners by means of mobile agent technology, the vast amount of heterogeneous information available on the Web actually becomes a obstacle against the right ones, which mainly because of the absence of a effective searching mechanisms and denotation of the online knowledge resources. Secondly, once the cooperation chain is set, how to integrate distributed information systems of other companies for synergy, i.e. on-demand integration, and afterwards detachment of the integrated systems remain the challenging issues [1].

To tackle the first-mentioned problem, we must be able to aggregate the relevant knowledge from the heterogeneous knowledge resources that require semantically enabled technologies to efficiently and precisely discover. At the same time the emerging technologies of Service Oriented Architecture (SOA) to a large extent enhance the interoperability of distributed heterogeneous systems[2], and the Semantic Web (SW) technologies make information and knowledge machine accessible and understandable rather than displaying them for interpretation by human. In addition, the emerging Grid technologies foster the coordination and sharing of computing, application, and data, storage, and network resources in a distributed environment [3]. Therefore, the solution for On-Demand integration of dynamic supply chain application may come from an approach to combine Semantic Grid (SG) and SOA technologies into the development of supply chain management.

2. EMERGING TECHNOLOGIES

The key to achieve the automation relies mainly on solutions to three issues: (1) how to make Web services of supply chain interoperable both syntactically and semantically; (2) how to automatically discover, based on the syntactic and semantic descriptions, the most appropriate information and services; and (3) assemble them to build the composite service. The emerging technologies of SOA and Semantic Grid along with its related technologies are briefly introduced in the following.

2.1 SOA

A SOA is ‘a set of components which can be invoked, and whose interface description can be published and discovered over the network’ [4]. Services are network addressable entities with a well defined, easy-to-use and standardized interface. Services like components are loosely coupled and often designed independently from the context in which they are used and composed. SOA typically involve multiple organizations that interact with networked systems where there is no single designer having full knowledge, control and ownership. In this sense, services are more coarse-grained than components.

Within a SOA, all applications are in the form of Web services including system functions for monitoring and controlling activities. Each service can invoke other services through shared communication protocols, infrastructures and interface specifications. It offers an ideal environment for applications integration.

Syntactical interoperability of Web services is achieved mainly using two common Web service standards: Web service description language (WSDL) and simple object access protocol (SOAP). WSDL is used to describe a Web service in terms of its interfaces and SOAP formalizes the XML-based message transportation between Web services.

2.2 Semantic Grid

The WS approach of SOA only resolves syntactic heterogeneities. For semantic heterogeneity, Semantic-Web-based technologies are required.

The Semantic Grid addresses the challenges in grid computing and applications by adding meaning through Semantic Web technologies (like ontology, annotation and negotiation process) to the grid [5]. Ontology as the core technology of SW is a formal, explicit specification of a conceptualization that provides a common vocabulary for a knowledge domain and defines the meaning of the terms and the relations between them[6]. Semantic heterogeneity can be handled dynamically by providing ontology mapping, merging, and versioning of ontologies through Semantic Web Services (SWS). SWS is WS semantically annotated based on ontology. The most widespread standard language for the Semantic Web is Web Ontology Language (OWL). Furthermore, the Semantic Web services initiative (SWSI) introduces OWL-

S as the representative technology for describing the semantics of individual Web services. OWL-S can be used to specify the semantics of the exchanged data, the functionality (through the reference to some service classification outside OWL-S), pre/post conditions, and other aspects of a Web service. These explicit specifications make the capabilities of individual Web services machine understandable so that automated Web service discovery is possible.

In this way, the semantic grid not only provides a general semantic-based computational network infrastructure, but a rich, seamless collection of intelligent, knowledge-based services for enabling the management and sharing of complex resources and reasoning mechanisms .

3. REQUIREMENTS FOR ON-DEMAND INTEGRATION

The integration issue in DSC management arises as the collaboration among enterprises becomes more and more frequent. In traditional methods, for companies adopting respective systems programmed by different source codes, in order to offer clients a complete set of services or to integrate the existing business applications (such as EAI and B2Bi) that are based on different architecture, communication protocols and data format, the inchoate E-business technologies such as EDI, Web EDI, are intended as joint solutions for complicated applications.

The SOA is well fit in terms of DSC applications. In this architecture a new mechanism for data interchange is proposed. An enterprise only provides and publishes a standard data searcher component where other partners can acquire it without providing different programs to different partners. Any one that uses company's core datum shall search from the database and package it in the format that its own system can understand. Taking the DSC system as an example for composition of a series of services, SOA aims at solving the dynamic problem of integrations in the means of adding, modifying, deleting services. In a DSC, every company can be a service provider and a service requestor at one time, forming a services sharing system consisting of different kinds of services.

For a sophisticated task, we may have to dynamically configure a collection of appropriate services including raw material stocking, transportation, wholesaling, retailing, etc., and automatically compose and execute them. In many cases, the constituent services are in a distributed heterogeneous environment, which may be geographically dispersed. Moreover, some constituent services may involve utilization of high computational power and large amounts of data. To address these issues, a list of requirements on DSC management solutions for On-Demand Knowledge is as following:

SOA/WS compliant: Adopt the SOA and Web Service standard such as WSDL, UDDI and SOAP to describe, publish/register and invoke knowledge services to increase their reusability and interoperability. Non service-oriented existing knowledge services, e.g. legacy KS, should be wrapped as WS.

Semantic Web based: WS should be semantically annotated base on ontology as SWS to facilitate matchmaking and service discovery.

SG enabled: Non Grid-enabled knowledge services which involve utilization of high computational power and huge amounts of data should be semantically annotated and wrapped as SGS.

Hybrid Reasoning: Ontology reasoning should incorporate with inference services in supply chain domain specific knowledge-based systems to facilitate automated service composition and execution.

Semantic Service Discovery and Negotiation: Use Semantic Web technologies and hybrid reasoning to enable automatic service discovery by matchmaking between SW-based services and the goals which is specified by the cooperating program via client orders.

Semantic Service composition and execution: Hybrid reasoning, BPM and choreography technologies should be incorporated together to enable dynamic configuration and automatic composition and execution of a collection of appropriated DSC services, and to ensure compatible choreographies between them.

4. CONCEPTUAL ARCHITECTURE FOR SG-ENABLED SOA APPROACH TO DSC

To meet the aforementioned requirements for On-Demand Integration, a generic conceptual architecture for Semantic Grid enabled SOA approach to DSC management is proposed and shown in Figure 2. The conceptual architecture

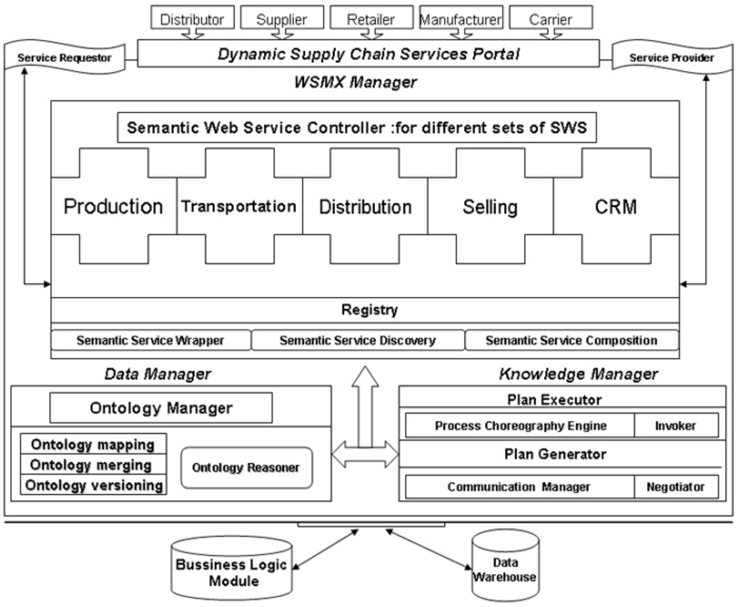

Figure 2. **Conceptual Architecture for SG-enabled SOA Approach to DSC**

encompasses four parts: *the Dynamic Supply Chain Services Portal, the WSMX*[7] *Manager, the Knowledge Manager* and *the Data Manager.*

The Dynamic Supply Chain Services Portal is provided as an entrance point to facilitate the process of searching for and utilizing the collection of SWS/SGS registered and published in the WSMX Manager.

The WSMX Manager contains basic several categories of supply chain services including production, transportation, distribution, selling, CRM and so on, which are through SWS Controller semantically annotated and wrapped as SWS or SGS. The manager may contain additional un-annotated WS or GS for temporary storage or for future semanticization. The Registry allows the Service Provider to register and publish its SWS/SGS through the portal. This Manager is a key part of the architecture that is responsible for the registry of SWS/SGS and automatic service discovery, composition and publication.

The Knowledge Manager plays the roles of Plan Generator (PG) and Plan Executor (PE) [8]. PG consisting of Communication Manager (CM) and Negotiator, is to communicate with Business Logic Modules and negotiate with Data Manager in order to be able to generate a desired optimized cooperation program of the supply chain into the service configuration. PE consisting of Process Choreography Engine (PCE) and Service Invoker, is to implement the program conveyed from PG by deploying and employing collection of SWS/SGS. PCE is a special container in charge of the sequence to initiate SWS/SGS and consequently defining a specific business process corresponded to SWS/SGS components. Container-provided functions enable long-running process executions that can even span enterprise boundaries, survive planned and unplanned outages, and facilitate Business-to-Business (B2B) collaboration. The Invoker is responsible for the actual invocation of the Web Services based on a standard API.

The Data Manager mainly emphasizes its maintenance on ontologies, among which mapping, merging and versioning are the most basic operations. Mapping could provide a common layer from which several ontologies could be accessed and therefore could exchange information in semantically sound manners[9]. Merging is to solve the problem that more and more ontologies are overlapping with the development of Semantic Web. Versioning is about the ability to manage ontology changes and their effects by creating and maintaining different variants of the ontology [10]. The Ontology Reasoner encompasses DL and Rule Reasoners to check the consistency and taxonomy.

5. RUN-TIME DEMONSTRATION WITH APPLICATION SCENARIO

In the context of Figure 1 supplied earlier, Figure 3 further illustrates driven by client *OrderA*, how a dynamic supply chain can respond to dynamic needs in an integrated information system environment. Manufacturer1 (M1) takes Supplier1 (S1) and Supplier3 (S3) as its main raw material provider, Supplier2 as candidate. M1 sends Material Order (MO) to S1 and S3, S1 intends to corporate with M1 whereas S3 does not. S1 then informs M1 of its stock situation, if stock is in shortage, M1 also can conveniently send MO to S2 to seek for a corporation. After the product is finished, M1 sends to a Product Inventory (PI) to its distributor candidates and distributor. Once the intent of the corporation confirms, a new real-time supply *ChainA* comes into being.

To support this Business Process (BP), we create three types of ontology: Entity Ontology, BP Ontology and Execution Ontology.

Entity Ontology contains different roles of the business world of the supply chain sector covering manufacturer, distributor, retailer, transporter, supplier, etc. Correspondingly the data and information exchanged among them such as product inventory, order, quotation, etc, serve as the foundation of communication and are important parts of entity ontology.

BP Ontology is used to describe the workflow of different BPs, reflecting business logic. It demands an effective abstraction, storage, re-utilization of the real-time BP knowledge.

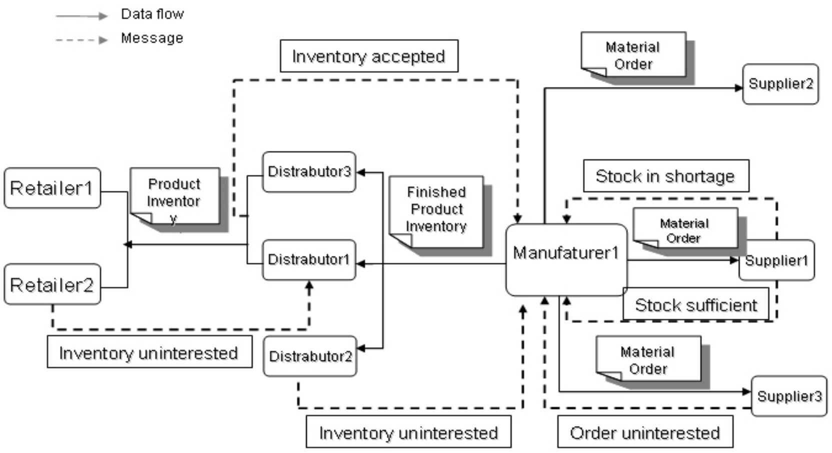

Figure 3. The Chain: A Formation Illustration

Execution Ontology is relevant to some specific activity bounded to the different roles of supply chain. Normally these activities can either change the internal or external management state, such as delivery of the stock control, or choose the decision-making methods such as transportation modes, lines and tools.

In this application scenario, *M1, S1, S2, S3, D1* (Distrabutor1), *D2, D3, R1* (Retailer1), *R2, MO, PI* are classified as Entity Ontology. Table1 shows a section of the owl file of the MO ontology.

Table 1. MO Ontology Example

```
<owl:Class rdf:ID=”Material Order”>

  <rdfs:subClassOf rdf:resource=”&Raw Material;Order”/>

      <owl:ObjectProperty rdf:ID=”hasMO_ID”/>
```

```
     <owl:ObjectProperty rdf:ID=”sent to”>
       <rdfs:domain rdf:resource=”#Order”/>
       <rdfs:range  rdf:resource=”Supplier”/>
      <owl:/ObjectProperty>
         ……………..
         <owl:Restriction>
       <owl:onProperty rdf:resource=”#sended to”/>
    <owl:minCardinality
       rdf:datatype=”&xsd;nonNegativeInteger”>2
       </owl:minCardinality>
        </owl:Restriction>
</owl:Class>
```

The above code shows that MO has to be sent to the type of “Supplier”, and at the same time being sent to at least to two suppliers. This constraint makes the non-suppliers’ systems automatically skip to process the MO by checking the restriction rdfs:range rdf:resource=”Supplier” if receiving the unexpected orders by mistake.

The business logic rule for cooperation of this application in the form of BP Ontology is given in Table 2. The business rules specify the conditions and consequences for business tasks to proceed.

Table 2. Example Business Rule to Deal with MO

RuleID	R001
Event	Received MO from M1(the trigger)
Type	Order Processing
Priority	1
State	Running
Condition	Null
Action	Confirmation of accepting MO or Not
Exception	Handle Exception Message
Pre-Action	Analyzing the MO ontology

In the WSMX architecture, Table 3 shows how the respective WSDL to describe the above service. With the use of Semantic Web markup languages data structures passed through Web Service interfaces are expressed by ontologies.

Table 3. **WSDL Service Description**

```
<?xml version=”1.0” encoding=”UTFi8”?>

<!ii Namespace Defintion ii> [...]

<message name=”OrderSent”>

<part name=”hasSupplier” type=”xsd:date”/>

<part name=”hasManufacturer” type=”xsd: string ”/>

</message>

<message name=” ”>

<part name=”hasRule” type=”xsd: Integer ”/>

</message>

<portType name=”MO_Processing”>

<operation name=”GetMO_Detail”>

<input message=”tns:OrderSent”/>

<output message=”tns:OrderRespone”/>

</operation>

</portType>
```

Once the confirmation of cooperation from supplier sent back to the Manufaturer, the order details can be monitored at any time.

6. SUMMARY AND CONCLUSIONS

The proposed framework for ontology development has research and practical implications. From the research perspective, we cannot claim that the proposed ideas are final and complete since research on DSC problems is continually evolving. On the contrary, new research can be carried out at a more detailed level, such as how problem identification can be accomplished or how ontology can be linked with data repositories. These and other findings can bring the proof of the proposed methodology, its revision and enhancement. From a practical perspective, SWS applications can be built based on the proposed framework to implement ontology-based IS for a DSC or for any other enterprise domain.

ACKNOWLEDGEMENTS

This research was supported by the Peking University Luen Tai center for Supply Chain System R&D.

REFERENCES

1. Z. Hai, Special Section: Semantic Grid and Knowledge Grid, *Future Generation Computer Systems.* Volume 23, Number 2, pp.281-282, (2007).
2. C.J. Yao, An Industry View on Service-oriented Architecture and Web Services, in *Proceedings - SOSE 2005: IEEE International Workshop on Service-Oriented System Engineering* (2005), pp.59.
3. C.D. Goble and D. Roure, The Grid: An application of the Semantic Web, *SIGMOD Record.* Volume 12, Number 2, pp.65-70, (2002).
4. T. Kaihara, Multi-agent Based Supply Chain Modeling with Dynamic Environment, *International Journal of Production Economics*. Volume 2, Number 22, pp.263-269, (2003).
5. M. Zaremba and E. Oren, *WSMX Execution Semantics*, WSMX Working Draft D13.2 v0.2 (2005). http://www.wsmo.org/2005/d13/d13.2/v0.2/20050202/ (Accessed October 2, 2006)
6. D. Booth, H. Haas, and A. Brown, *Web Services Glossary, Technical report,* World Wide Web Consortium (W3C) (2004). http:// www.w3.org/TR/ws-gloss/ (Accessed March 23, 2007)
7. W. Dai, *Collaborative Real-Time Information Services via Portals* (Idea Group Publication, 2007)
8. K. Yannis and S. Marco, *Ontology Mapping:The State of The Art* (2004), http://drops.dagstuhl.de/opus/volltexte/2005/40/ (Accessed June 2, 2006)
9. T. Gruber, *What is an Ontology?* (2002) http://www-ksl.stanford.edu/kst/what-is-an-ontology.html (Accessed June 3, 2006)
10. M. Klein and D. Fensel, *Ontology Versioning on the Semantic Web* (2005) http://www.cs.vu.nl/~mcaklein/presentations/2001-07-31-SWWS-Stanford.pdf