Integration Agent Negotiation and Data Global Consistency forms Automatic and None Bullwhip Effect Supply Chain

CHING-SHUN HSIEH[1] JUI-WEN HUNG[2]
[1]Department Commercial Technology and Management
[2]Department of Information and Management
[12]Ling Tung University
No.1, Lingdong Rd., Nantun District, Taichung City 408, Taiwan (R.O.C.)
TAIWAN (R.O.C.)
[1]sjsoon@mail.ltu.edu.tw
[2]harng@mail.ltu.edu.tw

Abstract: - Supply chain management (SCM) aims to efficiently integrate suppliers, manufacturers, warehouses, and retailers, not merely to ensure that merchandise is produced and distributed in the appropriate quantities, to the right locations, and at the right time, but also to minimize system wide costs while satisfying customer requirements. With global transnational enterprise layout trend, data consistency convergence study is a key of improving competitions and Bullwhip Effect problem in the GSCM. Besides, data consistency issues might induce different dependency relations in different extranet applications and partial dependency to several distributed original data source (ODS). In this paper, we consider the problem and solve it using a service routable consistency framework (Distributed Heterogeneous Web Service Routing based Portal; DHWSRP). The model based on a scalable routing service algorithm that dynamic reconfiguration forwarding data path within hierarchical enterprise region portals. By ripple propagate updating; SCM copies that existed in heterogeneity of enterprise database with specified ODS partially dependency relationships can be automatically updated. About efficient integrate SCM partner, we propose a Dynamic Information Exchange Center (DIEC) for creating a dynamic supply chain network in an Internet environment. Agent technology supports users in negotiating with upstream suppliers or downstream demanders and making decisions regarding partner selection. The framework allows enterprises to find more opportunities to cooperate with other partners. In business, such a framework not only reduces purchase costs and saves time for enterprises in reaching agreement but also eliminate Bullwhip Effect problem in GSCM.

Key-Words :- SCM, Intelligent agent, Negotiation, Outranking methods, Bullwhip Effect, Global consistency, Ripple service routing

1. Introduction

The computer and Internet revolution has changed traditional commercial activities, such as shopping, brokerage, negotiating, and retailing [10]. Many enterprises recognize that closed collaboration and rapid information exchange among supply chain members is essential for performing the purchase task in electronic trade environments. Figure 1, illustrates that the traditional environment of supply chain and information transfer is hierarchical. The information flow runs upstream, but the physical flow runs downstream.

Most issues in SCM involve discussing how to reduce inventory and satisfy customer needs. Enterprises also wish to improve their competitiveness by combining new information technology and supply chain management theory. The Internet has become widespread in business-to-business commerce, and enables businesses to communicate rapidly and efficiently. However, despite Internet technology being able to assist enterprises in communicating quickly and conveniently, enterprise delegates still need to meet face-to-face to discuss specific business deals. Software agents are personalized, continuously running, and semiautonomous [12][16][13]. Such agents can communicate with other agents through the Internet and perform specific tasks required by the user.

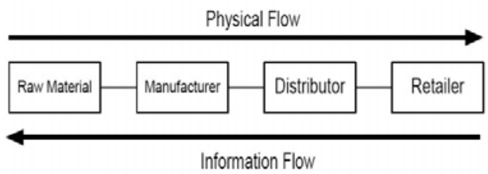

Fig. 1. Traditional supply chain

Generally, issues being negotiated are multi-faceted, and for example can include price, quantity, specification, quality, and so on. These attributes must be considered while negotiating process. Most critical situations require experts, known as representatives, to conduct this negotiation. Nevertheless, not all negotiation issues are so complicated that they require negotiator to handle those. Examples of such routine negotiation issues involve negotiations over products like paper, pens, printer ink, and so on.

This paper focuses on purchasing situations that are not directly related to the core business of a firm. An agent is used as a representative to negotiate with other enterprises in accordance with predetermined strategies. Utility function is used to assess opponent proposals and make counter proposals.

With global transnational enterprise layout trend increase, global replicas consistency convergence study is a key of improving competitions and Bullwhip Effect problem in the GSCM. Beside, the consistency issues might induce replicas dependency relations in different extranet applications and partial dependency to several distributed original data source (ODS). In section 5, we consider the problem and solve it using a service routable consistency framework (Distributed Heterogeneous Web Service Routing based Portal; DHWSRP). The model based on a scalable routing service algorithm that dynamic reconfiguration forwarding data path within hierarchical enterprise region portals. By ripple propagate updating service; SCM copy that existed in heterogeneity of enterprise database with specified ODS partially dependency relationships can be automatically updated.

Several studies are currently focusing on the design and concept of using caching, middleware and database replication technologies [23-26] to meet the requirements. However, existing consistency architectures do not meet automatically and globally consistency requirements. A novel global dependent replicas updating platform was proposed in first step, which consists of active, real time, automation, and global Forwarding Portal Clusters (FPC) web services routing technologies. When original patch data source (OPDS) object was triggered by ODS update event, all replicas with the specific OPDS dependent relationship could be automatically updated. The second step propose a adaptive update routing policies, and value added services based on usage mining results of replicas with specific OPDS dependency relationships.

1.1 Contributions of the paper

This paper focuses on how to build a dynamic supply chain through agent negotiation. The negotiation mechanism and outranking methods assist agents in gathering quantification and non-quantification conditions for selecting collaboration partners. We think this paper makes three key contributions, as follows: (1). Increase information transparency. (2). Uniform negotiation model. (3). Consider quantification and non-quantification criteria. This approach can save purchase time and costs. Additionally, a dynamic supply chain is formed to satisfy enterprise demand, to prevent it being frustrated by a single supplier running out of stock.

The text also presents a service routable consistency framework (DHWSRP) that is capable of supporting ripple propagate replicas updating copies that existed in distributed heterogeneity of storage. Besides, an individual object might induce different dependency relations in different applications and partially dependency to several distributed OPDS. By using a new scalable web service routing policies that can dynamic reconfiguration distributed forwarding patch data path in hierarchical enterprises storage and improving replicas data real-time correctness.

On the basis of the ubiquitous service routable framework, several analyses were built. The OPDS dependency spatial replicas usage mining rate is a function of dependency replica operation response time and turnaround time. The updating route policy scheduling rate is a function of usage mining and reservation priority. The production rate of activity represented as a time variable. Different value added update service schema was also analyzed. Apply scenario can really reduce the Bullwhip Effect and minimum the global patch turnaround time.

1.2 Outline of paper

The remainder of the paper is organized as follows. Related works are given in Section 2. In Section 3, we describe dynamic supply chain network and DIEC framework. The negotiation and outrank is explained with illustrative examples shown in Section 4. In Section 5, the forwarding Enterprise Portal (EP) agent [18-21] use pipeline mechanism automatically conversion all kinds of dependency patch and update special locality copies with OPDS dependencies. The OPDS hierarchical ripple propagation web service [22-24] routing algorithm discussion is given in Sect. 6. Section 7 presents our outrank example results. Finally, we conclusion our current results with suggestions for future research in Section 8.

2. Related Works

In this section, we will briefly describe related works about the traditional SCM management, agent, negotiation, outranking methods and global consistent issues applied to forms dynamic and none Bullwhip Effect GSCM.

2.1 Supply chain management

Supply chains in modern enterprises comprise a world-wide network of suppliers, factories, warehouses, distribution centers and retailers, through which raw materials are acquired, transformed into products, delivered to customers. To operate efficiently, supply chain functions must work in a tightly coordinated manner [1]. David [6] noted that supply chain integration is difficult for two primary reasons:

— Different supply chain facilities may have different, possibly conflicting, objectives.

— Supply chains are dynamic systems that evolve over time.

Croom [4] gathered various relevant definitions of supply chain management. Finally, this investigation concludes by discussing some issues related to supply chain management. These issues are:

— Cost reduction: Enterprises typically strive to reduce production costs, including time, stock, human costs, and so on.

— Information sharing: Enterprises must obtain more information, and thus can design appropriate strategies for enterprise benefit.

— Cross-organization integration: Management in a global society no longer involves striving in isolation. Therefore, enterprises must conduct cross organization integration.

2.2 Intelligent agent

Software agents, which now are very widespread, were first used several years ago to filter information, match individuals with similar interests, and automate repetitive behavior [12]. Agents are computer systems with numerous important capabilities. These capabilities are summarized and described below:

— Autonomy [8]: Users do not need to indicate how to work. Agents can finish task independently.

— Adaptive [1]: Agents can judge environmental change and adjust their task actively. Such agents generally have domain knowledge, and thus can easily identify ideal solutions based on the real conditions.

— Anticipatory [16]: Agents should not only predict user information through task or interaction with environment but also should alert users and react appropriately.

— Cooperation [8]: Agents can communicate with other agents by completing specific tasks together.

— Trustworthy [16]: An agent should satisfy the needs of user by using reliable methods to get user's trust.

Agents can adapt to environmental changes dynamically, and can model variation of management behavior among different managers [17]. Therefore, this work proposes to construct, using agent technology, and a supply chain communication and negotiation mechanism for reducing purchase cost.

2.3 Negotiation

Negotiation is a form of decision-making in which two or more parties jointly consider various possibilities with the goal of reaching a consensus [15]. Negotiation is complex because multi-dimension criteria must be considered and action must be prompt.

Rosenschein and Zlotkin [15] proposed a monotonic concession protocol with the following rules:

— Negotiation proceeds in a series of rounds.

— In the first round, both agents simultaneously propose a deal from the negotiation set.

— If one of the agents finds that the deal proposed by the other is as good as or better than its own proposal, it will accept the proposal.

— If no agreement is reached, the negotiation proceeds to another round of simultaneous proposals.

— If neither agent makes a concession in some round, then the negotiation terminates, with the conflict deal.

Faratin [7] designed a negotiation model involving autonomous agents. This multilateral negotiation was based on two parties who exerted a mutual influence on one another. This paper is based on this concept and considers how to apply this model in supply chain member negotiation and dynamic supply chain formation.

2.4 Outranking method

Purchasing decisions for supplier selection is strategically important for companies. Borer [2] reviewed existing decision models and concluded that the following properties are worth considering.

- Number of criteria and their nature.
- Interrelatedness of decisions.
- Type of decision rule used.
- Number of decision makers.
- Various types of uncertainty.

The outranking method is well suited for making multiple criteria decisions with qualitative as well as quantitative attributes. Numerous outranking methods currently exist, that is AHP, PROMETHE, ELECTRE and so on. No methods are perfect. This paper proposes to use ELECTRE III [3] as a tool for partnership selection. This model is simple and easy to use because users only need to setup some thresholds. This method can prioritize solutions based on user preferences.

2.5 Replicas mutual consistency solutions

Today's global distributed replicas consistent solution have been proposed using data integration middleware ; Patiño-Martinez [25]; techniques like xml-based integration middleware, xml-based data integration platform and xml-powered integration middleware. All these data integration system that enables enterprises to rapidly build web services and applications that can query multiple, disparate data sources and provides a unified result.

If distributed replicas with partial dependency relationships with specific OPDS object keep mutual inconsistency, the following problems will be happened. 1. The Bullwhip Effect will be occurred in GSCM operation. 2. Enterprise storages can't make sure the newest version copies. 3. Enterprise knowledge base will occur horizontal information lacks harmony. 4. If OPDS site can't control its replicas updating flow and usage mining results, it will not be able to support different priority and value added updating services. 5. The dependent replicas usage mining results can't be sent to OPDS site to make adaptive updating route policies and value added updating services.

3. Dynamic Supply Chain Network

Liautaud and Hammond [11] designed a supply chain extranet for expanding supply chain functionality. This paper proposes a dynamic supply chain model according Liautaud and Hammond's idea, using agent technology to solve the complex integration problem of the dynamic supply chain network. This model should automate and simplify the manufacturing process, and moreover, should allow members at all levels of the supply chain to reach their goals.

3.1 Dynamic supply chain model

In Figure 2, we illustrate a dynamic supply chain model. Retailers, distributors, manufacturers, and raw materials and logistic vendors share information in an Internet environment. The Dynamic Information Exchange Center (DIEC) then gathered the demand information of each member. Suppliers from every level provide their supply conditions and negotiate in the meeting room, while potential buyers evaluate the solutions proposed by opponents and make decisions. Other vendors can gather market movement information and make appropriate production forecasts using the DIEC framework.

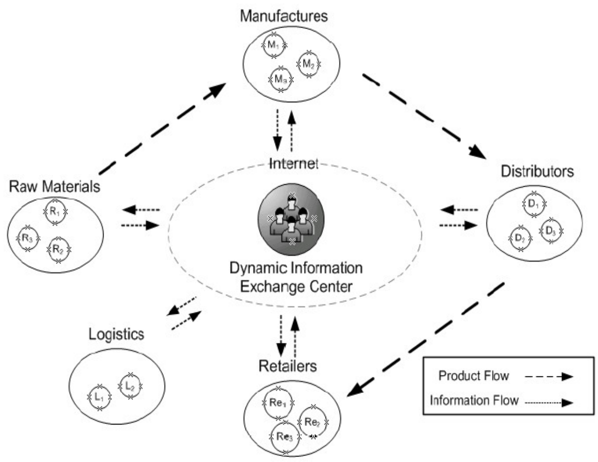

Fig. 2. Dynamic supply chain model

3.2 DIEC framework

The DIEC, depicted in Figure 3, divided into five functions: blackboard, meeting room, database, outranking model, and matchmaker. Any member of DIEC can be a demander. Members can request basic information from other enterprises or post their own needs on the blackboard. Other agents then can receive messages from the blackboard. Meeting rooms provide a private environment to support negotiations among agents with matching basic conditions.

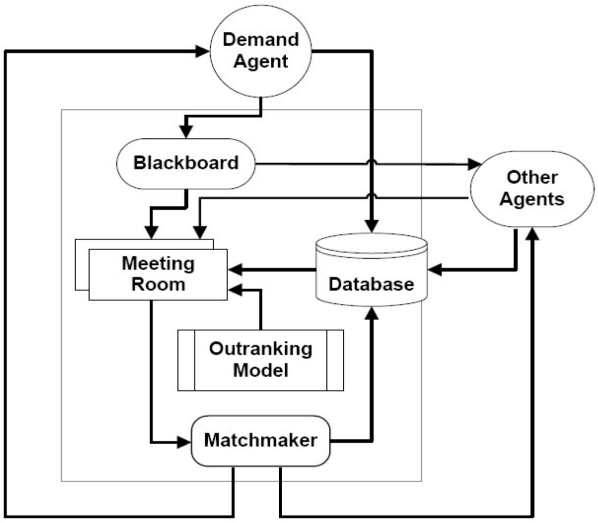

Fig. 3. DIEC Framework

After potential buyers post their requirements on the blackboard, they negotiate with potential suppliers. During negotiation, the demander agent can use the outranking model to rank solutions and select trading partners. Finally, the decision of the demander will pass to the matchmaker agent who announce the final supply chain network and record it in the database, and notify every member that joins the meeting room.

— Blackboard: Blackboard-based negotiation is derived from the mechanism of negotiation among intelligent agents using a blackboard as a media in which each agent exchanges information [9].

— Meeting room architecture: The meeting room provides a private space for both buyer and seller. The meeting room does not have a physical form. It is a serial of procedures. We will discuss at next session.

— Database: The database records the basic information of each agent, including location, business scope, sale items, and so on. The database provides other members with detailed information on the enterprise they are interested in.

— Outranking model: ELECTRE III uses the pseudo-criterion, with its indifference and preference thresholds, to explicitly make allowances for any imprecision/uncertainty in the data [17].

— Matchmaker: After supply chain formation, the contract is recorded in the database. The matchmaker agent then must confirm the inclusion of each new member in the chain and write into database.

4. Negotiate and Outrank

4.1 Sequence of demand, negotiation, and ranking

Figure 4 (arrow A) shows that demand flows upstream. Assume that demander agent W has potential suppliers M1 and M2. Furthermore, assume that M1 has potential suppliers Ua and Ub, while M2 has potential suppliers Uc and Ud. When W posts their demand on the blackboard, M1 and M2 can decide whether or not they want to trade. Before M1 proposes its supply solution, it must first check its stock. If M1 has insufficient stock to supply W, it will then post its own demand on the blackboard.

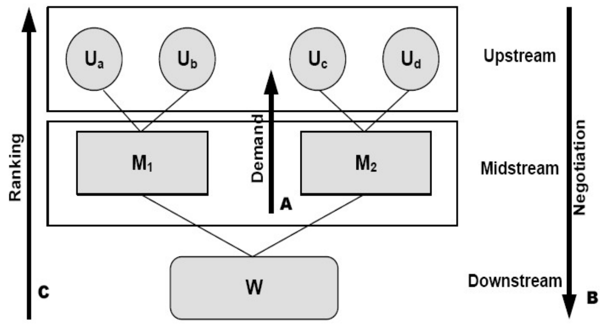

Fig. 4. Proposed sequence of demand, negotiation, and ranking

The negotiation sequence (arrow B) runs downstream. That is, upstream agents first negotiate with midstream agents to achieve agreement at first. Midstream agents then calculate their production costs and draft supply solutions, after which they make proposals to downstream agent. Supposing M1 has insufficient stock to provide W, and so must request support from upstream enterprises, otherwise, M1 will handle the demand of W by itself. Each level of negotiations, such as (Ua, Ub, M1), (Uc, Ud, M2) and (M1, M2, W), uses the same negotiation model to reach agreement. This model is also suitable for use in multi-tier frameworks. If Ua and Ub are interested in the demand of M1 demand and have sufficient stock to supply M1, then M1 will negotiate with Ua and Ub and seek a supply solution for its own supply problem before proposing its supply solution to W. If M1 has no consensus with Ua and Ub, it will lose the opportunity to negotiate with W. M2's activity is the same as M1.

Partner selection (arrow C) differs from the direction of negotiation. When M1 and M2 reach agreement with W, W has right to choose which one of these two suppliers will be its partner. M1 and M2 then use the same method to select their own upstream vendors.

4.2 Agent interaction diagram

Figure 5 describes the situation of agent interaction. Step 1 involves identifying possible supply chain choices. If no supply agent enters the room before the meeting room deadline, then the meeting room manager agent will cancel the meeting. The process proceeds directly to step 8.

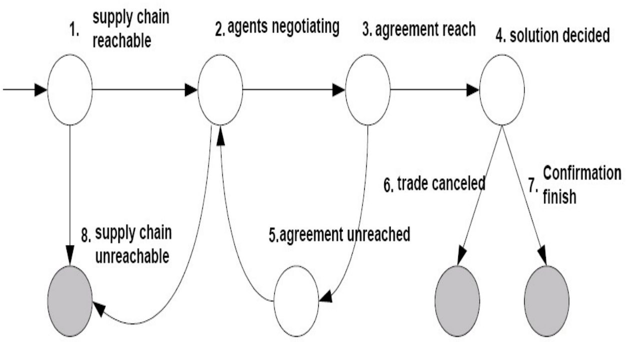

Fig. 5. Agent interaction diagram

Step 2 shows that some supplier agents are in the meeting room. Supply agents present their proposals to demand agents, and demand agents then assess those proposals. If the proposal of supply agent satisfies the needs of the demand agent, agreement will be reached. On the other hand, if the proposal does not satisfy the needs of the demand agent, the demand agent then presents a counter-proposal to the supply agent. These processes continue until agreement is reached, the counter-proposals of one or both parties are exhausted, or the allotted negotiation time is exhausted. Negotiation issues include quantity, price, due date, and so on. If no proposals can satisfy the requirements of the demand agent, the system skips to step 8. In this step, all supply agents negotiate with demand agent individually.

In step 3, the system gathers all of the proposals such that agreement is reached. Those supply agents will be candidates to form supply chain partner. The system can cancel any proposals that agents retract (step 5).

In step 4, the demand agent uses outranking method to select the cooperation partner. The outranking issues include supply agent proposal cost, credit condition, financial state, and so on. After the demand agent has made a decision, the system notifies the relevant supplier and other candidate suppliers. Once the relevant supplier has agreed to deal, the agreement becomes a contract and the meeting ends (step 7), else the deal will be cancelled (step 6).

5. DHWSRP Concept Model

We proposed novel global knowledge message consistency model - DHWSRP (Distributed Heterogeneous Web Service Routing based Portal) that can ripple propagate updating SCM copy that existed in internal heterogeneity of enterprise database with specified OPDS partially dependency relationships.

5.1 Enterprise portal model

Patch application replicas in intranet have three kinds of models. In Figure 6, patch replicas with partial specific remote OPDS dependency relationships were denoted by (a). Patch replicas with OPDS role in extranet enterprise flow chain was denoted by (b). A replica with partial dependency with a specific self-enterprise intranet OPDS was denoted by (c). Each EP can be played two roles: FP (Forwarding Portal) and FPC (Forwarding Portal Cluster).

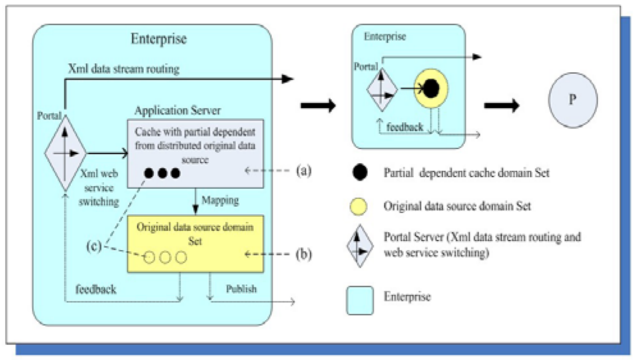

Fig. 6. Enterprise Portal (EP) Model

The ERP (Enterprise Root Portal) can get optimize and modify web service routing table information, according by receiving dependency replica updating request from low layer EPs. Then ERP configure each EP web service forwarding table based on a shortest path algorithm to create the appropriate distributed data path.

5.2 Problem formulation

For a given data path, a set of EPs are needed to perform web service processing functions from an ingress EP to an egress EP, and these EPs can be physically distributed and can be duplicated. Hence, several paths are possible between ingress and egress EP nodes since multiple duplicated EPs can perform the distributed data path. Assume a data path, DP is used to denote a vector of EPj involved in the data path formation:

$$DP = (EPj) \quad \text{for } 1 \leq j \leq F. \qquad \textbf{(1)}$$

which is also equivalent to a vector of sets A involved in the data path formation:

DP = (Aj)   for $1 \leq j \leq F$.   **(2)**

Figure 7 shows an example of a given FPC data path DP in enterprise intranet and extranet environment.

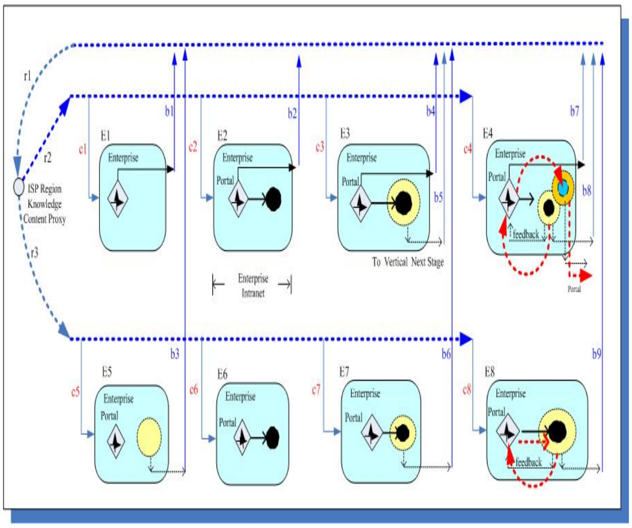

Fig. 7. A graph model of FPCs

Symbol description:
Enterprise is denoted by E. Routing path is denoted by r. flow path is denoted by c and b. or operation is denoted by |. EP encountered OPDS dependency patch replica flow will generate three kinds of message flow path , as follows formula.

Apply in enterprise extranet data path, like (3).
E5→b3→r1→r3→{c7|c8}→{E7|E8}→{b6|b9}→r1→r3

(3)

For redirect updating replica to other hierarchy enterprise, flow path like formula (4).
E5→b3→r1→r2→{c1|c2|c3|c4}→{E1| E2| E3| E4 }→{b1|b2|b4|b7}→r1→r2

(4)

For synchronize trigger multiple EP horizontal patch replica consistency, the feedback intranet OPDS updating copy must be built , like formula (5).
E8→b9→r1→{{r2 → same above (a)} | {r3 → same above (4)}}

(5)

5.3 OPDS ripple propagation patch routing mechanism

In this section, which was consists of OPDS subscribe and ripple propagation message chain shown in Figure 8. When OPDS object was triggered by update event, the message was propagation in active, real time, and automation. Then all patch replicas with the specific OPDS dependent relationship could be globally updated.

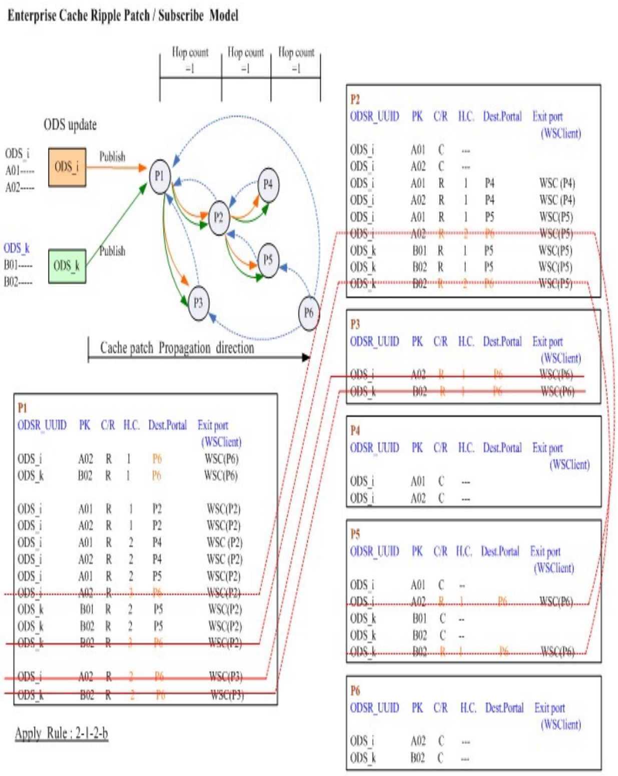

Fig. 8. OPDS ripple propagation patch chain

5.4 DHWSRP model of global patch replicas

In this paper, a novel framework was proposed which consists of active, real time, automation, and global routing patch replica from OPDS domain to DHWSRP set. See in Figure 9.

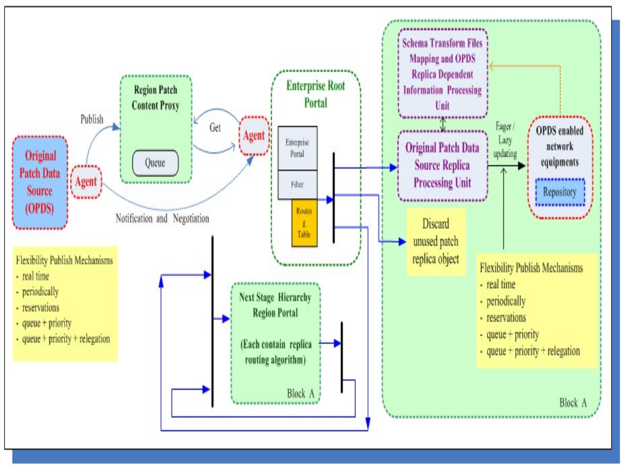

Fig. 9. DHWSRP concept model

When OPDS object occurred add/insert events.

According to subscribe lists, new copies will propagation push to all global EP which contained dependency replicas replica information. The EP then make one of the choices about discard/rerouting to lower hierarchy portal, automatic pipeline processing and schema transformation. All wide area OPDS dependent replicas replica consistency and correctness can be automatic maintenance. Enterprise intranet application system need not change any source code.

5.5 Use case diagram

In Figure 10 describe the UML use case diagram about the system (DHWSRP) concept model.

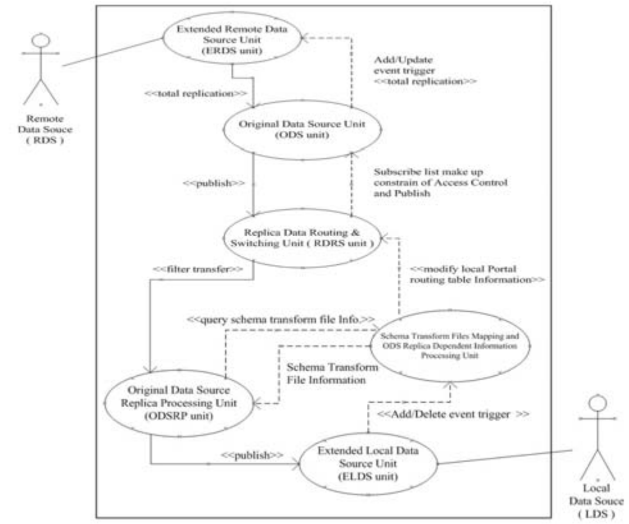

Fig. 10. System DHWSRP use case diagram

6. Ripple Propagation OPDS Routing Algorithm

Routing algorithm was built into each EP for assisting the xml patch stream route path choosing.

6.1 Basic principle

The routing table fields' description include: 1. OPDS dependency replica updates request information always maintained by EP. 2. Information included UUID value, PK value, Hop count, web service client etc. need to log in portal routing table from lower hierarchy enterprise request issues. 3. Portal's replica update routing information can come from intranet and extranet enterprise. 4. When a OPDS update replica publish to Portal, it will discard, send to processing unit, or reroute to lower portal depend on routing table check results. 5. For each route request from lower hierarchy layer portal will add a Hop Count (HC) value to routing table's HC value field.

Table 1. Routing table fields' definition

Routing Table (RT) fields	Description
action state flag (ASF)	denote when add/delete OKMS dependent replicas in enterprise LKS
OKMS template ID (UUID)	OKMS template identify defined by OKMS site
OKMS replica Primary key value(PK)	instance of specific OKMS template
local connect / remote connect (LC / RC)	dependency replica update request issued by local enterprise or lower level enterprise portal
Destination Portal ID (DPID)	original enterprise Identify that issues the update request
Hop Count value (HC)	the distance between the enterprise contained the newest routing information and request enterprise
WSC_port	call next stop portal web service

6.2 Routing algorithms

In Figure 11, apply the algorithm rules to each route request. The ASF (Action State Flag) is sent from lower layer EP by portal agent and web services. By Bellman-Ford Equation (DV algorithm) :

Let dx(y) is the minimum path cost from node x to y. Let c(x,v) is a moving path from x to v.

$$dx(y)=minv\{c(x,v)+dv(y)\} \qquad \textbf{(6)}$$

In distributed and asynchronous algorithm, each node will transfer its distance vector copy to all adjacencies at random time.

```
If (ASF==add) {
 // (rule.2-1) : ASF == add
  If (route request info.(UUID、PK、DPTD) sent from lower portal does not exist){
     //(rule.2-1-1)
     add this information into RT and upper cascade propagate this add event registry}
  Else If (route request info (WSC_port) identical){
       //(rule. 2-1-2-a)
       replace with the new route request info. and cascade propagate to upper portal when
       TTL trigger.}
  Else If (route request info (WSC_port) not identical){
      //(rule. 2-1-2-b)
      If (new HC value < RT's HC){
       //(rule. 2-1-2-b-I)
       replace with the new route request info. and cascade call back to lower hierarchy portal
       to disable the route with the same DPID when its WSC_port <> null }
      Else {
       //(rule. 2-1-2-b-II)
       Local portal do nothing then cascade call back to lower hierarchy portal to disable the
       route with the same DPID when its WSC_port <> null }

 }
Else {
// (rule.2-2) : ASF == delete
  If (route request info does not exist in RT){
   //(rule. 2-2-1)
   Local portal do nothing }
  Else {
   //(rule. 2-2-2)
   Find and delete the record in RT, and upper cascade propagate delete event registry }
}
Recursive()
```

Fig. 11. Replicas web services ripple patch routing algorithms

The following; Figure 12; is a demonstration for using above algorithm to adjust the P1 portal routing table.

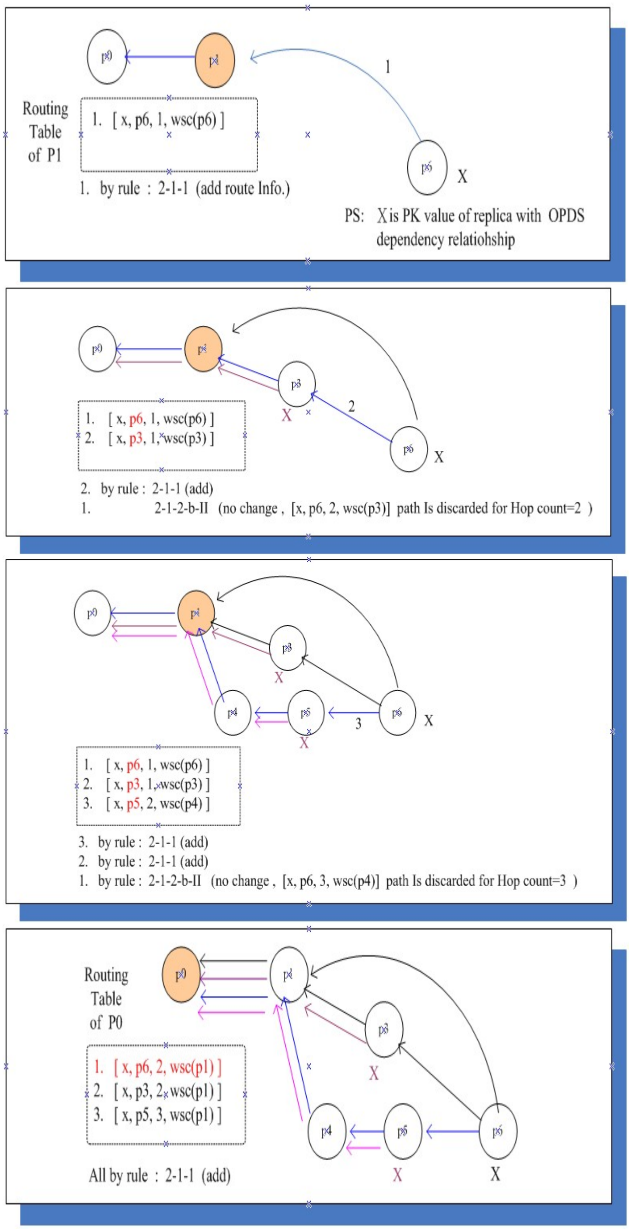

Fig. 12. Example for routing algorithm

Figure 13, 14, 15 is a demonstration for enterprise portal add/change/delete its position issues.

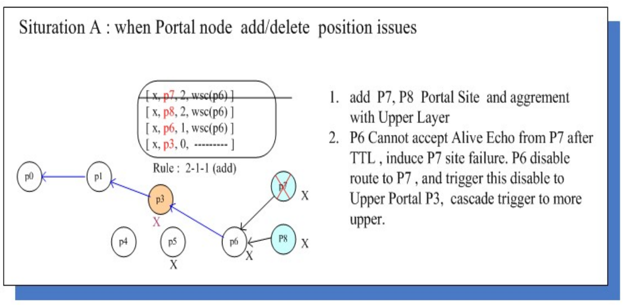

Fig. 13. Demonstration when portal node add/delete its position issues

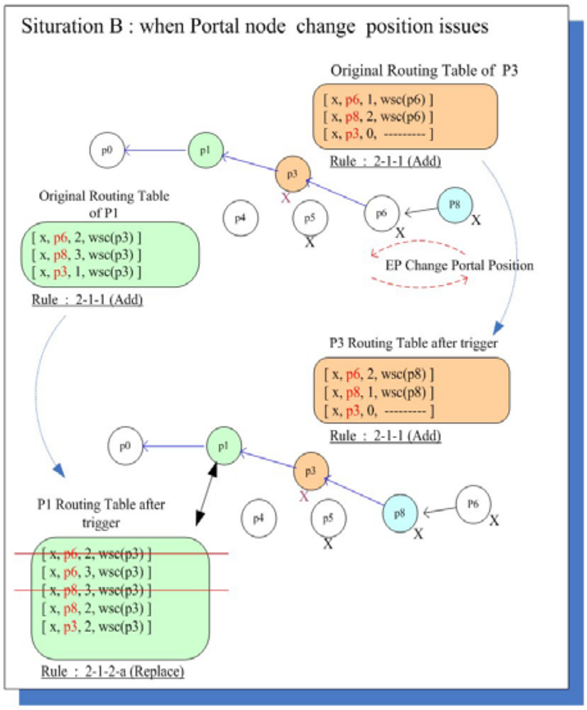

Fig. 14. Demonstration when portal node change its position issues

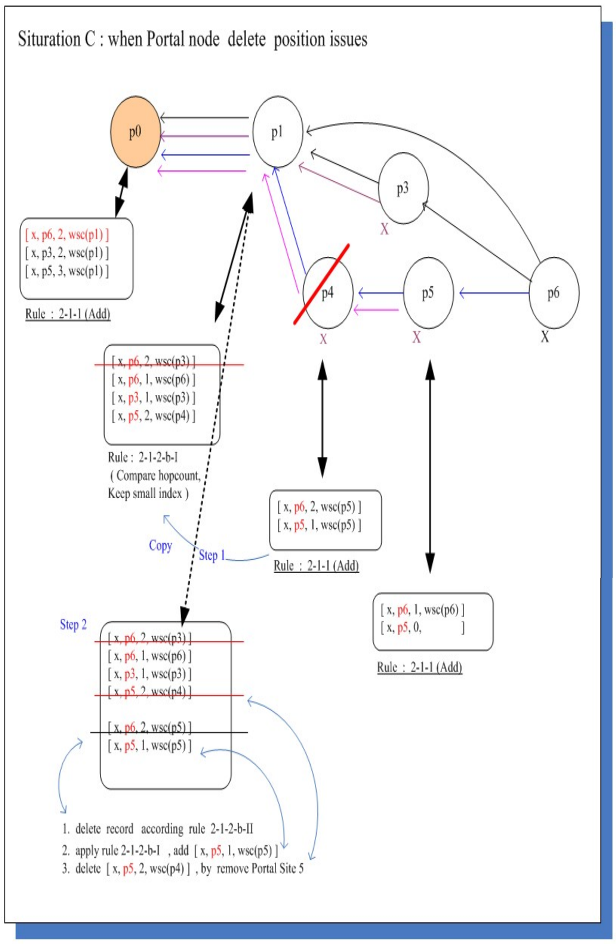

Fig. 15. Demonstration when portal node delete its position issues

6.3 Estimate of routing algorithms for distributed global consistency

In this section we estimate three kinds of distributed consistency schemas applied in hierarchical and distributed storage environment like schools. 1. P2P push based eager replica updating. 2. Proxy pull based lazy replica updating. 3. ODS push based and active web service propagate routing algorithm for global consistency.

6.3.1 Simulation goal

All administrative information on the hard disk containing the ODS copies of partial dependency set. Through a set of comprehensive, proactive and real-time update mechanism can achieve consistency of the global convergence goals. So that all the schools' decentralized system, will no longer have access to inconsistent or outdated information.

6.3.2 The scope of simulation

The scope of simulation is based on Ling Tung university campus administrative group, access the information on disk which was part of dependency with Personnel part ODS copy. At present, for improving the application performance by the latest copy, accuracy and timeliness toward 2 directions. Including:

1. Replicas of time and space to improve regional exchange of information on the frequency problem.
2. Accuracy and real-time upgrade distributed dependency copy.

The following will discuss 3 kinds' global consistency models for evaluating and comparing the effectiveness of its operations.

(a) P2P push based eager replica updating.

The uses of P2P subscribe/publish eager push mode send updated copy of the personnel changes to subscribed agency. The advantage is immediately, one-way dissemination to all registered administrative units. The shortcomings is the Personnel must to set up individual Peer to Peer's relations subscribe lists. Personnel must also be effective at any time to verify the transmission list of groups available. How to ensure to meet the goal of global convergence consistency is the most difficult. The operation is shown in Figure 16.

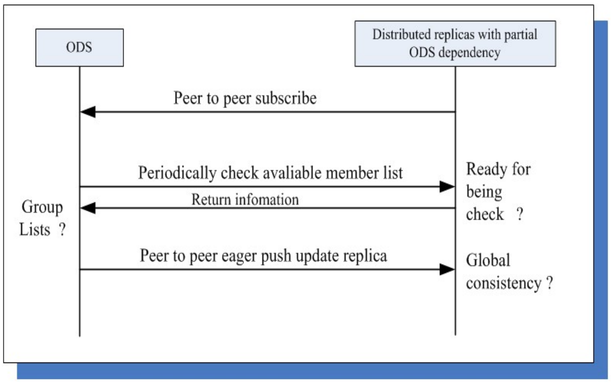

Fig. 16. P2P push based eager replica updating

(b) Proxy pull based lazy replica updating

Personnel sent the staff-to-date update copy to school proxy. If all administrative units of application required the ODS dependency copy must through pull and data integration mechanism way. The advantage of lazy changes is to reduce non-essential message conversion and the time to send a message. The disadvantage that proxy server is a single point of failure risks. That wills not real-time changes all the dependency replicas on the copy of the application easy access to information inconsistent results. The operation is shown in Figure 17.

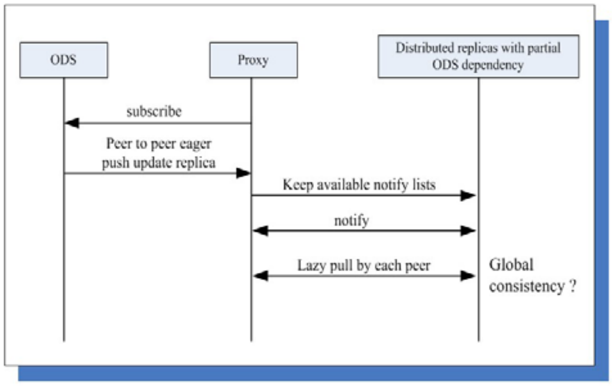

Fig. 17. Proxy pull based lazy replica updating

(c) ODS push based and active web service propagate routing algorithm

The goal of routing algorithm proposed is using Subscribe/Publish and active XML web service propagates routing Mechanism for global consistency. In a comprehensive, active and hierarchical automation ripple propagation way to reach global consistency of dependency information copy in school. All administrative units' application is able to integrate information at the lowest cost. Advantage: 1. No longer based on individual applications repeat investment in different database middleware system. 2. There is no need to periodically pull to obtain a copy of dependency-to-date information. For the shortcomings: 1. Root portal and business units to be implemented with replicas routing algorithm. 2. Enterprise portals have to implement ODS update replica schema transformation. The operation is shown in Figure 18.

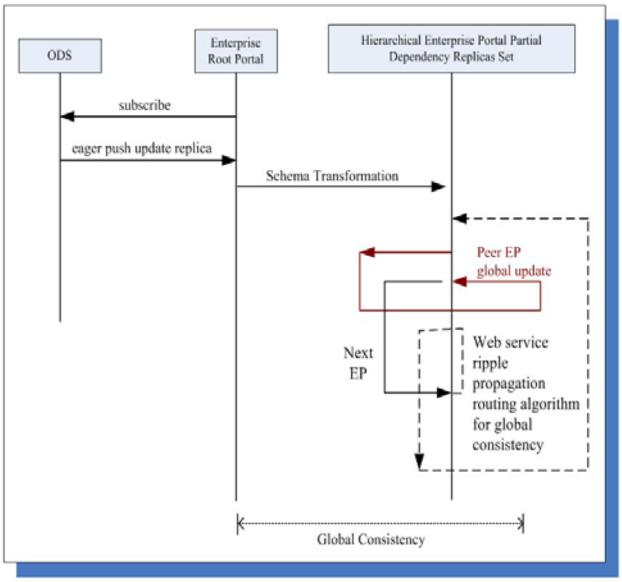

Fig. 18. ODS push based and active web service propagate routing algorithm

In Figure 19, assume in measure the global consistency convergence time is ignored the following metric including processing delay, queuing delay, transmission delay. There are considered the storage replica updating delay and network propagation delay. In practically, then transmission delay is usually between several μs. and ms.

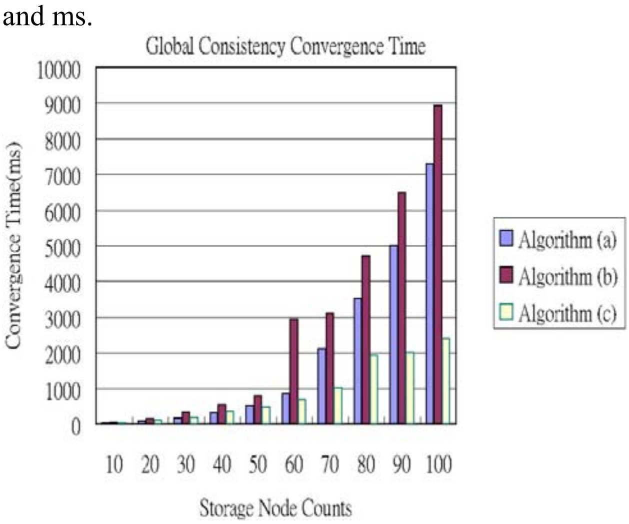

Fig. 19. Comparison (a)(b)(c) of different algorithms on global consistency convergence time

7. Outrank example

As for negotiation issues, we give two assume parameters: price and purchase quantity. Each agent has its own acceptance range for the alternative. For example, Table 1 is the price and quantity acceptance range of M1 and W.

Table 2. Acceptance range

	Price	Quantity
W	[15, 18]	[800, 1200]
M1	[16, 20]	[900, 1400]

The value function is used to calculate those acceptance ranges. As depicted in Figure 20, each agent calculates value by its utility function.

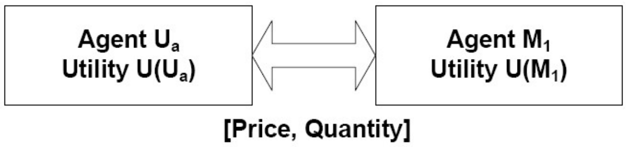

Fig. 20. Each agent calculates value by its utility function.

M1 wishes to negotiate with W. The acceptance range of W regarding price is from \$15 to \$18, denote P[15, 18]; while the acceptance range for quantity is from 800 to 1200, denote Q[800, 1200]. Assume the best benefit alternative for W is [15, 800]. M1's acceptance range for price and quantity is [16, 20] and [900, 1400] respectively. Assume the best benefit alternative for M1 is [20, 1400]. Furthermore, the preference weight of W for price and quantity, respectively, are 0.6 and 0.4; and the preference weight of M1 for price and quantity, respectively, are 0.4 and 0.6. According to the above base data, a value can be calculated for each round [price, quantity] alternative using the value function. The calculated result is called the utility value. For example, assume one of the alternatives that M1 propose is [17, 900]. This alternative can be calculated as a value for W, as follows:

$$\frac{(17-15)}{(18-15)}\times 0.6+\frac{(900-800)}{(1200-800)}\times 0.4=0.5 \quad \textbf{(8)}$$

Therefore, after calculating all possible alternatives, M1 can rank these alternatives from high to low. W also can rank its own alternatives. Both sides propose alternatives based on their own preference sequences. When M1 proposes an alternative to W, this alternative can be calculated as a utility value by W. If this utility value is less than the alternative W is planning to propose in the next round, W will not accept this alternative, and instead propose its own preference alternative to M1. M1 then calculates utility value of the alternative of W, and makes a decision. If the alternative utility value of W exceeds the utility value of that which M1 is planning to propose in the next round, then M1 will accept the proposal of W. In this case agreement is reached.

While M1 and M2 both reach agreement with W, W calculates the purchase costs of each supplier and establishes a performance table as Table 3. The credit means the degree that a vendor be trusted by other supply chain partner. Moreover, quality represents whether the state of the product is good or bad. Credit and quality are all established based on the recognition of W. If W prefers a particular enterprise, it will assign that enterprise a high degree of trust. These evaluations can be established prior to negotiation.

Table 3. Performance Table

	Cost	Credit (1~100)	Quality (1~100)
M_1	160,000	82	90
M_2	165,000	90	85

Table 3 illustrates that W finished negotiating with M1 and M2 and make a performance table. Table 4 provides three thresholds, namely an indifferent threshold, a preference threshold, and veto threshold, as well as the criteria weight. The three thresholds illustrate user's feelings regarding each item listed in the performance table.

Table 4. Three kinds of thresholds and criteria weights

	Cost	Credit	Quality
Indifferent Threshold	2,000	2	3
Preference Threshold	5,000	5	5
Veto Threshold	10,000	10	10
Weight	0.5	0.3	0.2

According to the above data, W ranks those two suppliers by using the ELECTRE III outranking method. The result is that M2 becomes more suitable for W than M1. This information is received by the matchmaker, and the matchmaker then notifies each of the members involved in the negotiations and confirms the agreement.

8. Conclusions and Future Work

This paper focuses on how to build a dynamic supply chain through agent negotiation. The negotiation mechanism and outranking methods assist agents in gathering quantification and non-quantification conditions for selecting collaboration partners. During the negotiation process, the agent serves as a tool that estimates opponent supply conditions based on user presetting of their value function. Meanwhile, value function is a mathematical formula that represents the preferences of each participant in different negotiation conditions. ELECTRE III can sort candidates according to demand preferences. The demander need only set their preference, indifference, and veto thresholds, and can use these parameters to sort potential suppliers. This automatic mechanism not only reduces purchase costs but also saves time for enterprises in reaching agreements.

We think this paper makes three key contributions, as follows:

－Increase information transparency

－Uniform negotiation model

－ Consider quantification and non-quantification criteria

In practice, this model is suitable for automating the task of work-related material purchasing. Enterprise negotiators only need to consider primary raw materials purchasing, and can let agents handle the purchase of other work-related materials. This approach can save purchase time and costs. Additionally, a dynamic supply chain is formed to satisfy enterprise demand, to prevent it being frustrated by a single supplier running out of stock.

In this paper, we consider the problem and solve it using a service routable consistency framework (Distributed Heterogeneous Web Service Routing based Portal; DHWSRP). The model based on a scalable routing service algorithm that dynamic reconfiguration forwarding data path within hierarchical enterprise region portals. By ripple propagate updating; SCM copy that existed in internal heterogeneity of enterprise database with specified ODS partially dependency relationships can be automatically updated. It's mainly contribution as follows. 1. The ODS dependency replicas usage mining rate is a function of dependency replica operation response time and turnaround time. 2. The updating route policy is a function of usage mining and reservation priority. The result can expand wide area enterprise database replicas contained ODS dependency relationships with real time global consistency maintenance abilities. Besides, we propose a Dynamic Information Exchange Center (DIEC) for creating a dynamic supply chain network in an Internet environment. Agent technology supports users in negotiating with upstream suppliers or downstream demanders and making decisions regarding partner selection. The framework allows enterprises to find more opportunities to cooperate with other partners. In business, such a framework not only reduces purchase costs and saves time for enterprises in reaching agreement but also eliminate Bullwhip Effect problem in GSCM.

References:

[1] Barbuceaunu, M., Teigen, R., and Fox, M.S.: Agent Based Design and Simulation of Supply Chain Systems, *WET-ICE '97, IEEE Computer Society Press*, 1997, pp. 36-41

[2] Borer, L.D., Wegen, L.V.D., and Telgen, J.: Outranking Methods in Support of Supplier Selection, *European Journal of Purchasing and Supply Management* , 1998, Vol.4., pp. 109-118

[3] Buchanan, J., Phillip, S., and Daniel, V., Project Ranking Using ELECTRE III, Department of Management Systems University of Waikato, Hamilton, New Zealand, *Research Report Series,* 1999.

[4] Croom, S., Romano, P., and Giannakis, M.: Supply Chain Management: An Analytical Framework for Critical Literature Review, *European Journal of Purchasing and Supply Management*, 2000, Vol. 6., pp. 67-83

[5] David, P. and Peter, S.: Predictive Planning for Supply Chain Management, *Proceedings of the International Conference on Automated Planning and Scheduling*, Cumbria, UK, 2006.

[6] David, S.L., Philip, K., and Edith, S.L.: Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies, *McGraw-Hill*, 2000.

[7] Faratin, P., Sierra, C., and Jennings, N.R.: Negotiation Decision Functions for Autonomous Agents, *International Journal of Robotics and Autonomous Systems*, 1998, Vol. 24. pp. 159-182.

[8] Genesereth, M.R. and Ketchpel, S.P.: Software Agents, *Communication of the ACM*, 1994, Vol. 37., pp. 48-53

[9] Ito, T., and Salleh, M.R.: A Blackboard-Based Negotiation for Collaborative Supply Chain System, *Journal of Materials Processing Technology*, 2000, Vol. 107., pp. 398-403

[10] Kraus, S.: Strategic Negotiation in Multiagent Environments, *MIT Press,* 2001.

[11] Liautaud, B. and Hammond, M.: e-Business Intelligence: Turing Information into Knowledge into Profit. *McGraw-Hill,* 2000, pp. 244-245

[12] Maes, P.: Agents that Reduce Work and Information Overload, *Communication of the ACM*, 1994, Vol. 37., pp. 31-40

[13] Minghua, H., Rogers, A., Xudong, L. and Jennings, N. R.: Designing a Successful Trading Agent for Supply Chain Management, *Proc. 5th Int. Conf. on Autonomous Agents and Multi-Agent Systems*, 2006.

[14] Rogers, M., Bruen, M., and Maystre, L.Y.: ELECTRE and Decision Support: Methods and Applications in Engineering and Infrastructure Investment, *Kluwer Academic*, 2000.

[15] Rosenschein, S.J. and Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation among Computers, *MIT Press*, 1994.

[16] Sycara, K., Decker, K., Pannu, A., Willliamson, M., and Zeng, D.: Distributed Intelligent Agents, *IEEE Expert,* 1996.

[17] Yung, S.K. and Yang, C.C.: Intelligent Multi-Agents for Supply Chain Management, *IEEE SMC'99 Conference Proceedings*, 1999., pp. 528-533

[18] Yung Bok Kim and Soon Woo Lee, Performance Evaluation of Mobile Agents for Knowledge-Based Web Information Services KES-AMSTA, *Springer Lecture Notes in Computer Science*, 2007, Vol. 4496, pp. 209-218.

[19] Ruey-Kei Chiu and Kuo-Chin Tsai and Chi-Ming Chang and S. C. Lenny Koh and Kuan-Chih Lin, "The implementation of an agile information delivery system in building service-oriented e-healthcare network", *Int. J. of Enterprise Network Management Inderscience Publishers*, 2007, Vol. 1, No.14., pp. 283-298.

[20] Ákos Hajnal and David Isern : Knowledge Driven Architecture for Home Care CEEMAS, *Springer Lecture Notes in Computer Science*, 2007, Vol. 4696,. pp. 173-182.

[21] Aurora Vizcaíno and Juan Pablo Soto and Javier Portillo-Rodríguez and Mario Piattini, A Multi-agent Model to Develop Knowledge Management Systems, HICSS, *IEEE Computer Society*, 2007., p. 203

[22] Mahmoud Brahimi and Mahmoud Boufaïda and Lionel Seinturier, Integrating Web Services within Cooperative Multi Agent Architecture, *AICT/ICIW, IEEE Computer Society*, 2006., p. 197

[23] Nicolas Singer and Jean-Marie Pecatte and Sylvie Trouilhet, Combining Web Services and Multi-Agent Technology to Increase Web Cooperative Capacities, *IEEE International Conference on Internet and Web Applications and Services (ICIW'07), Mauritius*, 13/05/07-19/05/07, electronic medium.

[24] Oh Byung Kwon, Multi-agent system approach to context-aware coordinated web services under general market mechanism , *Decision Support Systems*, 2006, Vol.41, No.2,. pp. 380-399

[25] M. Patiño-Martinez, R. Jimenez-Peris, B. Kemme, G. Alonso. Consistent Database Replication at the Middleware Level., *ACM Transactions on Computer Systems (TOCS). In Press,*. 2003

[26] Thanasis Loukopoulos, Ishfaq Ahmad, Static and Adaptive Data Replication Algorithms for Fast Information Access in Large Distributed Systems, *20th IEEE International Conference on Distributed Computing Systems (ICDCS'00),* 2000, p. 385