A Sophistication Framework for a Mother Company-Driven Global Manufacturing Network

Kwangho Park
Department of Business Administration, Hanyang University
(oobepark@hanyang.ac.kr)

The main purpose of this paper is to propose a sophistication framework for a global manufacturing network (GMN) driven by a mother company to autonomously propagate and coordinate transaction data that are exchanged among manufacturing partners. The framework is based on conceptual fundamentals of previous research that provide a step toward ultimate successful collaboration in the supply chain and employs mobile agents for the coordination and propagation of transaction data. Maintaining the integrity of transaction data linked to a huge information web is difficult. With the sophistication functionalities of this framework, it becomes easy to effectively control the overall GMN operations and to accomplish the intended goals. The current level of sophistication focuses on the transaction data propagation. The sophistication level may be expanded up to business intelligence in the future.

Received : December 24, 2008 Revision : March 06, 2009 Accepted : March 10, 2009 Corresponding author : Kwangho Park

1. Introduction

There has been a lot of speculation regarding the practicability of supply chain management (SCM). It would be foolish to wish away the organizational realities and attempt to develop the integrated systems because there are bound to be implementation failures (Dearden, 1972; Premkumar, 2000; Moberg et al., 2003). Major barriers to SCM are as follows: a lack of understanding of what supply chain collaboration actually implies; the context for collaboration, in terms of when to collaborate and with whom; and ignorance of culture, trust, information exchange, and supply chain wide performance measures. Thus, it is proposed that a segmented supply chain approach limiting collaboration to a small but potentially critical number of partners is a more appropriate context for such collaboration (Barratt, 2004). Recently, global manufacturing enterprises such as Samsung Electronics and Hyundai Motors have started to build their own global manufacturing network (hereafter GMN). Ideally, GMNs are expected to be able to overcome critical failure factors of SCM such as inherent complexity, over-reliance on technology, a lack of trust, and a failure to differentiate with whom to collaborate (Ireland and Bruce, 2000; Mentzer et al., 2000; McCarthy and Golocic, 2002; Sabath and Fontanella, 2002; Moberg et al., 2003; Baratt, 2004). In a GMN, a powerful and dominant global manufacturing enterprise called a "mother company" drives tight ERP-to-ERP (Enterprise Resource Planning) integrations initiative on its globally distributed suppliers while excluding downstream distributors. Guaranteeing fair and long-term financial security for suppliers, a mother company is expected to lead the connected companies to successful SCM implementation.

Even if a GMN may be able to remedy all inherent problems, a fundamental issue remains: how to maintain the integrity of transaction data linked to a huge information web. Errors of any participant are likely to propagate to at least the adjacent and possibly all subsequent participants in either direction (Sproul and Kiesler, 1991). Without carefully designed standards and structures, the level of uncertainty can become worse due to adverse symptoms such as the bullwhip effect that distorts demand information as it is transmitted up the chain (Lee et al., 1997). Small mistakes may end up breaking up the entire trust chain even if ideal information sharing can be accomplished by strong enforcement by a mother company. In short, these issues and problems boil down to informational and planning gaps that exist when deploying the outcome of such collaborative efforts (Andraski, 1998). Thus, the challenge is to determine how to improve the process and information flow using existing technology.

This paper proposes a sophistication framework, SMART,[1] for GMNs to autonomously propagate and coordinate transaction data that are exchanged among manufacturing partners. Especially, the framework reveals in depth the propagation aspect of the transactional information flows between organizations: how information is coordinated and integrated. As stated in Shar et al. (2002), the strategic implications of the framework are as follows: if companies are coordinated more with trading partners and their information systems are more highly integrated, they generally improve on each of the performance variables.

Admitting that the implementation of any SCM framework is almost impossible not only because of technical problems but because of economic and socio-political problems (Kumar and van Dissel, 1996; Premkumar, 2000; Moberg et al., 2003), we construct the architecture that provide a step toward the ultimately successful collaboration in the supply chain based on conceptual fundamentals of previous research. SMART employs mobile agents for the coordination and propagation of transaction data. Various integration aspects such as transaction data integration and propagation, information sharing, and trans-

1) Abbreviation for Sophistication Master for Autonomous pRopagation of Transaction data.

action cost are investigated and incorporated into the framework.

2. SMART framework

In GMNs, transactions take place in every company. The transactions create a great amount of information that is closely inter-related. Transaction data flows in both directions. Orders from a mother company to the first-tier suppliers are continuously propagated back toward raw material suppliers. In addition, parts are then delivered back toward the mother company. Upon delivery, accounts payable and accounts receivable must be settled firstly and journalized. Finally, payments flow backward as orders. Within each company, transaction data flows through internal process chains and, at some points, both inward (i.e. receiving) and outward (i.e. order and payment) transaction data is generated and sent to other companies. The whole GMN is comprised of a set of inter-related transaction data networks, each of which represents the formal operation model of the individual company.

SMART should rely on an adequate operational system such as ERP of the internal processes of each company (Ptak, 2000). SMART can be conceived as a set of ERP systems each of which controls internal operations (transactions) and among which various interactions take place. Transaction data is inter-related and processes are tightly integrated. As Bingi et al. (1999) summarize, however, tight process integration can propagate mistakes made in one department of a company into other departments in other companies and back to itself in real time. Also, the original mistake can be easily magnified as it flows through the GMN transaction data network. Thus, companies must be aware of the potential risk of error and take proper steps, such as monitoring the transactions and taking immediate steps to correct the problems. They must also have a formal plan of action describing the steps to be taken if errors are detected. A proper line of communication must exist so that all affected trading partners can be notified as soon as the errors are detected.

2.1 Architecture

In a GMN, the mother company takes the initiative to connect the ERP systems of most partners into its own ERP system. The ERP integration proceeds to upstream partners gradually until the entire manufacturing processes of a GMN are completely integrated. Tools that let companies establish ERP-to-ERP links are application integration suites, e-commerce systems, and middleware. An integrated ERP (I-ERP) network system links trading partners together so best practices emerge and the total effort is enhanced. Arrayed around the nucleus ERP network are the types of collaborative relationships that can improve a value chain constellation (Poirier and Bauer, 2000).

ERP vendors such as SAP and Oracle have been trying to extend the functionality limit of their products to tie them together into one package as many disparate applications, including order entry and manufacturing, supply chain management for transportation and warehousing, and sales force automation for customer service (Green, 2001).

The Internet has three key attributes that impact strategy that can be exploited in business-to-business dealings: real-time information, the ability to build and unleash the power of relationships, and the positive effect it has on the economics of software (O'Keeffe, 2001). Recently, a few industry consortiums, i.e., Microsoft's value chain initiative (VCI) and SAP's adaptive supply chain networks, are trying to implement the idea that enables a single-point-of-contact concept through the Internet as well as to evolve the enterprise system of the future. In a GMN, the mother company builds its own private trading network on the web. For example, GLONETS is an e-commerce system that enables real-time e-procurement transactions between Samsung Electronics and its parts vendors on the Internet. Such a SCM interface provides an easy portal to the ERP system of the mother company.

As shown in <Figure 1>, SMART consists of a fundamental data integration model (FDIM), information sharing model (ISM), transaction data network model (TDNM), and transaction propagation model (TPM). The following sections will describe each model. Each model employs a set of agents that are autonomous and independent (Michael and Jennings, 1995). As proposed by Liu et al. (2005), the architecture expands WSCM which abstracts the workflow-supported inner supply chain management system. However, SMART's primary focus is on the interaction aspects of transaction data propagation.

The supply chain council[2] developed and endorsed the supply chain operations reference (SCOR) model. The five distinct processes of the SCOR model 7.0 are plan (P), source (S), make (M), deliver (D), and return (R). Here, source, deliver, and return are external transactions that relate associated trading partners. A GMN is established to best comply with hierarchical processes in SCOR model 7.0. Especially, SMART corresponds to "enable" SCOR process type that prepares, maintains, or manages information or relationships on which planning and execution processes rely. On the other hand, "planning" and "execution" SCOR process types are excluded in our research scope.

2.2 The fundamental data integration model

Transaction data, such as order, delivery, accounts receivable/payable settlement, refund, payment, and return registered in a partner in a GMN, is supposed to be propagated to the adjacent partner through an adapter based on an XML interface in an ERP-to-ERP integration environment. However, in order to fully trace how the propagations take place in a GMN, the fundamental data integration model (FDIM) is required with the following agents:

2) Supply Chain Council, Inc., 303 Freeport Road, Pittsburgh, PA 15215, www.supply-chain.org.

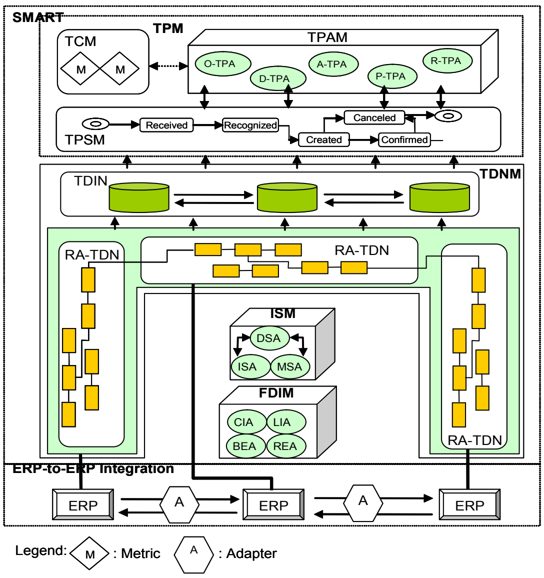
- Code interface agent (CIA): Part codes are integrated such that a proper interface module takes care of any differences in code identification.
- Language interface agent (LIA): By adopting Unicode, any language interpretation or presentation is properly taken care of.
- BOM (Bill of Materials) explosion agent (BEA): BOM of a final product is expanded to include BOMs of any child parts or subassemblies.
- Routing explosion agent (REA): Routing of a final product is expanded to include routings of any child parts are included.

2.3 The information sharing model

SMART assumes that information must be shared among trading partners in real-time. Ross (1998) illustrates the various flows in any SCM by summarizing pervious findings as shown in <Table 1>. Based on FDIM, the information sharing model (ISM) is composed with three type of information sharing agents which provide the real-time execution status of a GMN for planning:

- Inventory status agent (ISA): inventory levels of materials, parts, and final products
- Manufacturing status agent (MSA): manufacturing status of an ordered product
- Delivery status agent (DSA): global delivery estimation and status for an ordered product

Internally, DSA employs ISA and MSA to make delivery estimations which is the most critical activities in planning. The mother company receives an order for a product and finds that the inventory level is not enough to deliver immediately. Then, DSA is called to estimate the Available To Promise (ATP) estimation. ISA and MSA provide the current status of inventory and manufacturing and compute best delivery dates based on standard lead times that are predetermined by contracts among the mother company and suppliers in a GMN. More sophisticated planning algorithms are available in supply chain planning (SCP) software (Cowdrick, 1995) and may be incorporated in designing DSA. Since planning is beyond the scope of the paper at the present, optimal estimations will be attempted in future study.



<Figure 1> Architecture of SMART

<Table 1> Flows in SMART: abstracted from Ross(1998), p. 172

Flow	Description
Information	Marketing information about new and current customers, advertising, pricing, competitors, and other forces in the marketplace
Title	Ownership
Promotion	Information concerning prices, marketing sales materials, advertising, delivery, and other elements
Negotiation	Information concerning agreement on price and other sales terms
Ordering	The actual placement of customer and inventory replenishment orders
Financing	Capital that must be acquired and allocated to finance the transfer of ownership
Risk	Risks on inventories that can become obsolete, damaged, or unsalable
Payment	Cash payment

2.4 The transaction data network model

The company's ability to create, collect, assimilate, access, and transfer transaction data must be in alignment with the velocity of the activities necessary to effectively execute the management of supplier, customer service, manufacturing, logistics, and financial functions. SMART is based on the integrated processes that put companies in touch with the whole, with one another, with customers, and with suppliers in ever-changing patterns of relationships (Savage, 1996).

We can define two types of transaction data:

- Internal transaction data created by internal events such as WorkOrder, DeliveryRequest, Inspection, etc.
- External transaction data created by external events and delivered to other trading partners such as SalesOrder, Shipping, PurchaseOrder, OutsourcingOrder, Payment, Return, etc.

When external transactions occur among partners in a GMN, the resulting transaction data are recorded in the associated companies. These transaction data can be inter-related with each other, for example, each sales order entry refers to purchase order entry, each receiving entry refers to a delivery entry, etc. Such inter-relationships are established not only within individual company but also cross two participating companies. Thus, two types of the transaction data network (TDN) can be defined:

- intra-transaction data network (RA-TDN): A network of transaction data that are intra-related within a company
- inter-transaction data network (ER-TDN): A network of transaction data that are inter-related across companies

Then, TDN can be defined as an ER-TDN within which a number of RA-TDNs are interrelated with each other. Four basic types of relations are defined in terms of the cardinalities of transaction relationships: separate (one-to-one); consolidated (many-to-one); split (one-to-many); and split and consolidated (many-to-many). The more complex cardinalities make TDN more difficult to manage.

Having defined TDN, we now can get to the instantiation of transactions. By the definition of TDN, actual instances of transaction data occur in each company participating in a GMN. These transaction data instances comprise transaction data instance network (TDIN). When a transaction data instance is added to TDIN, it must be decided which preceding transaction data instance triggered the transaction and, therefore, the corresponding transaction data instance must be properly referred. Thus, the inter-company referential integrity constraints must be guaranteed as well as the intra-company referential integrity constraints. Assuming that every single transaction data instance is created due to preceding transaction data instances, a complete TDIN can be maintained and, therefore, provide a whole picture of activities in a GMN.

By TDIN, a complete trace network can be built both inside a company and across multiple companies for a search parameter, i.e., an order number, delivery number, etc. Each TDIN can be tracked by a primary key such as OrderNumber, DeliveryNumber, PaymentNumber, and ReturnNumber. Further details of TDIN traces are discussed in the next section. By nature, TDIN is a bi-directional network in a sense that transaction data instances occur in both directions, from the mother company to raw materials suppliers and vice versa. The trace is a complex task because it must pass borders of many companies.

2.5 The transaction propagation model

2.5.1 Transaction propagation

In practice, transactions do not always proceed as defined. TDN specifies, in fact, a normal execution flow of transactions. However, the real troubles are abnormal use cases (Jacobson et al., 1992). For the execution of any transaction, it must be anticipated that cancellations or corrections will take place. Abnormal use cases represent such modifications of the whole or a part of previous executions. However, the cancellations or corrections cannot be performed in an isolated fashion due to the nature of process integration. The effect of cancellations or corrections must be completely propagated through the TDIN. Improper executions for the abnormal use cases may result in violations of referential integrity constraints. Companies must be aware of the potential risks of the errors and take proper steps, such as monitoring the transactions and taking immediate steps to rectify the problems should they occur. They must also have a formal plan of action describing the steps to be taken if an error is detected. A proper means to communicate to all the parties who are victims of the errors as soon as the errors are detected is extremely important.

The transaction propagation model (TPM) represents the range and relationship of transactions that are triggered by an initial transaction.

The propagation can take place within a company or cross multiple companies. Also, the propagation, either normal or abnormal, gets more complicated depending on the cardinalities of relationships between transactions. For example, consider the example of a company that participated in a GMN. Suddenly experiencing a shortage of manufacturing materials, production managers noticed that it was due to incorrect bills of materials and made necessary adjustments. However, the company did not have any procedures to notify others of the error. Thus, first they have to cancel orders for the materials to correct the overflow of the material inventory. Then, the cancellation has to be propagated along TDIN back to raw material manufacturers. The domino effect of the errors had already affected other companies.

There are two types of normal transactions:

- Regular: This type of transactions takes place as pre-defined either by business rules or contracts.
- Rush: This type of transactions takes place against business rules or contracts. Thus, it must specify specific time and specifications such as urgent due date and excessive order amount.

Also, there are two types of abnormal transactions:

- Change with grace (CWG): This type of transactions can take place gracefully without causing financial problems to participating companies.
- Change with penalty (CWP): The propagation cannot take place without causing financial problems to participating companies. Thus, it may not be propagated as intended.

CWG abnormal transactions can be applied to suppliers before the decoupling points of the product routings. Because suppliers before decoupling points are make-to-stock manufacturing and therefore operate in a stable order environment, they can take the changes without serious financial loss. The temporary overflow of the material inventory will be soon consumed by future orders. However, suppliers on and after decoupling points are more vulnerable to changes because they are make-to-order or engineering-to-order manufacturing and have less room to adjust to those changes on pressed orders.

For the illustration of propagation complexity, we will focus on an extreme case of TPM. As shown in <Figure 2>, TPM involves various orders. If company A sends a purchase order (POrder) to company B, it becomes a sales order (SOrder) for company B. The SOrder in company B, internally, triggers POrders for parts or raw materials and outsourcing orders (OOrders). The remaining work orders (WOrders) are executed internally and thus isolated in company A. When the ordered products are ready to deliver in company B, delivery request transactions are registered and then the products are actually shipped and the corresponding transactions are registered.

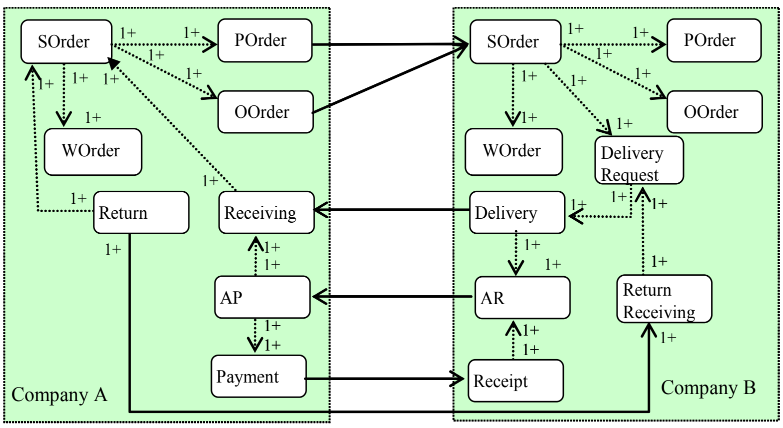
The delivery transactions are in turn propagated to company A as receiving transactions. In this extreme case, cardinalities of all internal transactions are assumed to be many-to-many. Then, we can imagine how a change or cancellation of the original POrder in company in A may affect the entire transaction propagation. The amount and the complexity of the propagation must overwhelm the responsible departments in each company.

The propagation does not always take place as defined as in TDN. This is because most companies operate on the basis of planning. As discussed in the leagility issue, companies try to postpone the final specification of their products. In a leagile company, transactions take place on the basis of planning. Before decoupling point, transactions are triggered by planning, and after that, transactions are triggered by external orders. Thus, an SOrder may not trigger purchase or outsourcing orders because the ordered products have been already produced or the parts may have been already purchased.

Therefore, the leagile companies in general have much simpler TDIN than non-leagile companies. As more transactions take place on the basis of planning, the propagation becomes less complicated. In a completely planned situation, all companies in SMART may exchange planned orders and thus generate only regular transactions. The more rush (normal or abnormal) transactions take place, the more complicated TDIN becomes.

2.5.2 The transaction processing state model

As transactions are propagated, various internal and external works flow in a GMN. Recently, the workflow aspect of SCM has been widely explored (van der Aalst, 2000; Liu et al., 2005). Thus, in SMART, the transaction processing state model (TPSM) can be modeled in order to understand the various aspects of workflows in a GMN (see <Figure 3>).



<Figure 2> An extreme case of condensed diagram of ER-TDN

In a GMN, the starting point of TDIN is a sales order transaction in a mother company. This sales order may generate internally purchase orders to the first-tier suppliers. In turn, upon the receipt of these purchase orders sent by the mother company, the first tier suppliers generate sales orders. These initial sales order transaction data must be approved, business rules associated with it checking: item and specification; unit price; quantity and delivery date; current inventory; normal lead time; and safety stock. Then, the actual sales order transaction data instance is created and recorded in the database.

As the transaction is propagated through the TDIN, the transaction data instance is partially referenced and then finally becomes completely referenced. For management purposes, it is assumed that a transaction data instance becomes "complete" when it is "completely referenced" by the subsequent transactoin data instances. Here, "completely referenced" implies that the transactoin data instance is referenced completely by any successor transactoin data instance so that no more direct and transitive references are possible.

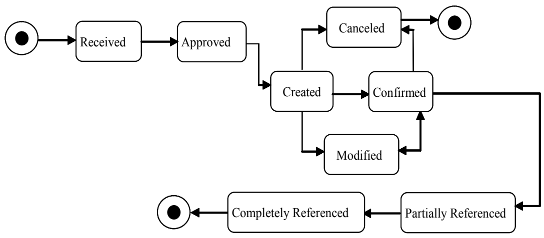
After creation, a transaction data instance may be canceled and modified. The cancellation leads it to the termination state. On the other hand, either created or confirmd-state transaction data instance can be modified and then confirmed again. Such abnormal use cases can make TDIN much more complicated.

2.5.3 The transaction cost model

The fundamental nature of enterprise information integration is thoroughly investigated in Giachetti (2004). The operational complexity in manufacturing is defined (Frizelle, 1998; Frizelle and Woodcock, 1995) and recently expanded to operational complexity between partners within a supply chain (Sivadasan et al., 2002). Park and Kusiak (2005) also defined the operational complexity of ERP operations. Based on the previous research, we defined the following GMN operational metrics as the transaction cost model (TCM):

- Number of "complete" transaction propagation (NCTP): the total number of transaction data propagation that has been completely referenced
- Number of "incomplete" transaction propagation (NITP): the total number of transaction data propagation that has not been completely referenced
- Average duration of complete transaction propagation (ADTR): the average duration of complete transaction propagation
- Maximum duration of complete transaction propagation (MDTR): the maximum duration of complete transaction propagation
- Also, we can define transaction cost of an order of a product (TRCO) as the total number of transaction data instances propagated throughout a GMN. At best, TRCO would be the total number of PurchaseOrders and SalesOrders for the parts at the first level of BOM explosion.

At the worst, TRCO would be the total number of PurchaseOrders and Sales-Orders for the parts at the every level of BOM explosion.



<Figure 3> Transaction propagation state model

2.5.4 The transaction propagation agent model

Mobile agents have provided the opportunity to reduce transaction costs for electronic markets, making for a more efficient allocation of goods and services in the economy (Malone et al., 1987). When executing, mobile agents move from one server to another via the network (Mandry et al., 2000). The agent's program stops at one server and sends its code, data and execution state to the target server where it will continue its work. The agent performs its tasks at the remote server(s), collects the results, and waits on the server. The components of mobile agents were initially proposed in Rothermel et al. (1997): execution state, program code and data.

Adopting the fundamental components of mobile agents, SMART employs the transaction propagation agent model (TPAM). TPAM consists of the following transaction propagation agents (TPAs):

- Order-TPA (O-TPA): It is responsible for the propagation of a purchase order of an ordering company to a sales order of a supplier company.
- Delivery-TPA (D-TPA): It is responsible for the propagation of a delivery of a supplier to a receiving of an ordering company.
- Account-TPA (A-TPA): It is responsible for the propagation of an account receivable of a supplier to an account payable of an ordering company.
- Payment-TPA (P-TPA): It is responsible for the propagation of a payment of an ordering company to a receipt of a supplier.
- Return-TPA (R-TPA): It is responsible for the propagation of a return of an ordering company to a return-receiving of a supplier.

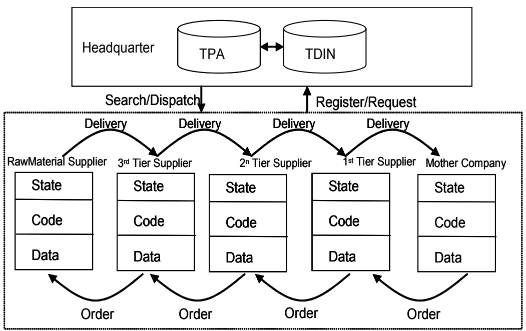
Each TPA should perform both normal (create and read operations) and abnormal (update and delete operations) transaction propagations. Thus, when TPA is dispatched, the calling transaction should designate an appropriate use case such as create, read, update, or delete. A condensed version with O-TPA and D-TPA is shown in <Figure 4>. When a mother company places an order of parts to the 1^{st} tier suppliers, the corresponding purchase order transaction data is registered in the ERP system of the mother company. Then, as the result of the order processing, an O-TPA is created and sent to the 1^{st} tier suppliers. As arriving at a 1st tier supplier, the O-TPA triggers the sales order transaction processing, creating sales order transaction data in the 1st tier supplier's ERP system. Later, the 1st tier supplier ships the parts to the mother company and the corresponding delivery transaction data is created. Upon the complete processing of the delivery transaction, a D-TPA is created and sent to the mother company. When the D-TPA arrives at the mother company, it triggers the receiving transaction processing. Such operations by O-TPA and D-TPA are repeatedly propagated toward the end of the upstream suppliers in a GMN.

2.5.5 The design of transaction propagation agent

Every time TPA is created, it is registered in SMART headquarter where agents are globally controlled and fundamental registry services are provided. Because mobile agents move along the GMN, information regarding the current execution state and location should be provided. With the registry services from SMART headquarter, each trading partner can request a particular agent with a primary key of the transaction. Once an agent lands at a company, it starts executing its code depending on the state. Thus, in SMART headquarter, two types of database are maintained: agents and TDIN.

The typical TPA can be defined as an enterprise java bean (EJB):

```
public class TPA implement EntityBean {
  public LocationID locationID; // State
public PropagationState
propagationState; // State
public TPA sender; // Data
public TPA receiver; // Data
public TPA forwarder; // Data
public DateTime sendingTime; // Data
public UseCaseType useCaseType; //
Data
public TransactionData transactionData;
// Data
public PrimaryKey primaryKey; // Data
public ForeignKey foreignKey; // Data
public Inter-CompanayForeignKey in-
ter-CompanyForeignKey; // Data
public TPA create(TransactionData)
throws RemoteException; // Code
public TDIN read(PrimaryKey) throws
RemoteException; // Code
public TDIN update(PrimaryKey,
UpdateList) throws RemoteException; //
Code
public void delete(PrimaryKey) throws
RemoteException; // Code
  private void sendACK(TPA) throws
```



<Figure 4> A condensed version of TPAM

```
RemoteException; // Code
}
```

Sometimes, a TPA propagates into multiple suppliers. For example, in a manufacturing company where several parts or raw materials must be purchased in order to produce a product and fulfill the corresponding order, the associated TPA must be sent to several suppliers. Thus, the agent propagation gives rise to a network of agents' interactions. Also, the data interdependency information regarding the referential relationships with the primary key and foreign key must be recorded in the agent registry database in the SMART headquarter. In order to incorporate the inter-company referential relationship, the typical referential integrity constraint is expanded as follows:

Primary_Company.Primary_Table.Primary_Key⇔Foreign_Company.Foreign_Table.Foreign_Key

The following is a sample scenario for which company A registers a purchase order and the purchase order is propagated to company B by an O-TPA (see <Figure 5>):

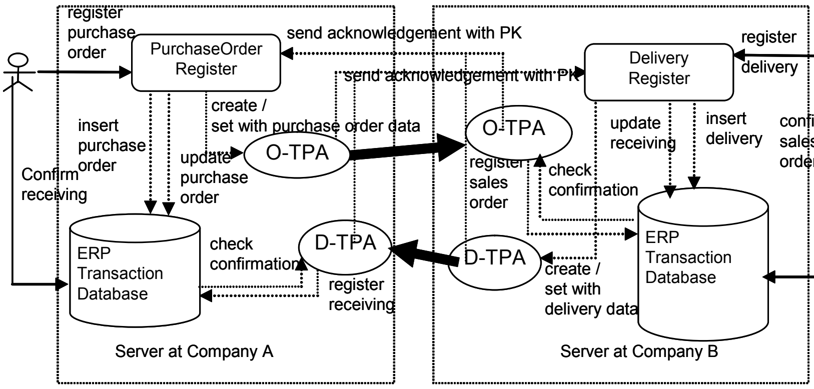
- A user of company A registers a purchase order into database of company A.
- An O-TPA for the propagation is created with the purchase order information and the useCaseType attribute is set to "create."
- Before sending the O-TPA to company B, SMART headquarters creates a TDIN with the purchase order number as the primary key.
- The O-TPA is sent to company B which was specified in in the supplier atribute of the purchase order (this becomes the forwarder of O-TPA).
- As soon as the O-TPA successfully arrives at the server of company B, it changes the state attribute to "received."
- The O-TPA checks the business rules for sales orders and set the state attribute to "approved" if all of the business rules are satisfied. Otherwise, it waits for a sales manager's approval.
- The O-TPA registers a sales order transaction with the purchase order information into database of company B and changes the state attribute to "created."
- Before sending the acknowledge to company B, SMART headquarter updates the TDIN with the sales order number.
- The O-TPA sends company A an acknowledgement with the value of primary key of the sales order (salesOrderNumber) of company B.
- The server of company A receives the acknowledgement and updates salesOrderNumber attribute (inter-Company Foreign Key) of the corresponding purchase order with the salesOrderNumber of company B.

Now, when the ordered parts are ready to ship in company B, a delivery transaction is registered and propagated to company A by a D-TPA as follows (also see <Figure 5>):

- A user of company B registers delivery information into database of company B.
- A D-TPA for the propagation is created with the delivery information and the UseCaseType attribute to "create" is set.
- The D-TPA searches the correspsonding O-DTA with salesOrderNumber in TPA database and sets the state attribute of the O-DPA to "partially referenced" or "completely referenced" depending on whether the ordered quantity is delivered all or partially.
- Before sending the D-TPA to company A, SMART headquarter updates the TDIN with the delivery number.
- The D-TPA is sent to company A which was specified in the customer atribute of the sales order (actually it is the sender of the corresponding O-TPA).
- As soon as the D-TPA successfully arrives at the server of company A, it changes the state attribute to "received."
- The D-TPA checks the delivery for the business rules and set the state attribute to "approved" if all of the business rules are satisfied. Otherwise, it waits for a logistics manager's approval.
- The D-TPA registers a receiving transaction with the delivery transaction information into database of company B and sets the state attribute to "created."
- Before sending the acknowledge to company B, SMART headquarter update the TDIN with the receiving number.
- The D-TPA sends company B an acknowledgement with the value of primary key of the receiving (receivingNumber) of company A.
- The server of company B receives the acknowledgement and updates receivingNumber attribute (inter-company foreign key) of the corresponding delivery with the receivingNumber of company A.

Now consider a read operation of TPA. When a GMN administrator would like to see how a purchase order has proceed at the moment, O-TPA.read(PurchaseSalesNumber) is called. The following pseudo code illustrates how the read operation is executed:

```
O-TPA.read (PurchaseSalesNumber) {
  Initialize TransactionPropagationSet = NIL;
  Set ForeignKeySet =
  Inter-CompanyForeignKey(PurchaseSale
  sNumber);
  if ForeignKeySet is NIL
  { return TransactionPropagationSet; }
  else { While NOT ForeignKeySet is NIL
    Set ForeignKey.NextForeignKeySet =
    D-TPA.read(ForeignKey);
    Add Foreign Key to
    TransactionPropagationSet; }
}
```



<Figure 5> Transaction propagation agent execution between companies.

Finally, consider an update operation of TPA. When a GMN administrator would like to change the quantity of an item in a purchase order, O-TPA.update(PurchaseSalesNumber, ItemNumber, Quantity) is called. The following pseudo code illustrates how the update operation is executed:

```
O-TPA.update (PurchaseSalesNumber,
ItemNumber, Quantity) {
  Initialize Status = CWG; //
  CHANGEWITHGRACE
  set ForeignKeySet =
  Inter-CompanyForeignKey(PurchaseSale
  sNumber);
  if ForeignKeySet is empty
  { return Status; }
  else { While NOT ForeignKeySet is
  empty AND Status = CWG
    Set Status = D-TPA.update(Foreign
    Key, ItemNumber, Quantity); };
   }
```

The operational mechanism of mobile agents can be applied to any transaction because the topology of a GMN contains repeating patterns. The remaining transactions such as A-TPA, P-TPA, and R-TPA are sophisticatedly propagated by the same scenarios.

2.5.6 Implementation

Each model is implemented by Web services. Web services are enabled by two technologies (Gisolfi, 2001): Firstly, eXtensible Markup Language (XML) is the technology for moving transaction data across the Web. And, SOAP (Simple Object Access Protocol) uses XML messages to invoke remote methods. Working hand-in-glove with XML and SOAP, two other technologies enable service discovery: Firstly, UDDI (Universal Discovery Description and Integration) is a specification for information registries of Web services. And, Web Services Description Language (WSDL) describes functions provided by a given Web service. Thus, communications between models are enabled by SOAP with XML documents. Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web (Tidwell, 2000). Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.

The agreement between companies consists of the ontology mapping between the mother company and its suppliers. The skeleton of such agreement is designed in Liu et al. (2005) as a format of exchanged messages encoded in XML including service entry point, service parameter (primary key), and security requirement.

3. Conclusions and Suggestions for Further Research

Samsung Electronics (SEC) launched an ERP-to-ERP integration initiative in late 2004 and supported its 257 1st tier suppliers to adopt ERP by the first quarter of 2006. SEC selected five local ERP solutions for the suppliers to integrate with its SAP system through global logistics network system (GLONETS)-an EDI system (www.secbuy.com). Eventually, GLONETS will be replaced by a virtual GMN site where SAP of SEC and various ERP systems of suppliers are integrated completely. SamsungSDS which is the systems management company for SEC, is participating in the project.

SamsungSDS is embedding some of SMART-like functionalities into its ERP solution, *uni-ERPII Special*. At the moment, purchase, logistics, and payment transaction data are automatically propagated by an XML adapter applying propagation business rules and agents. As more suppliers join GLONETS integration initiative, more sophisticated transaction coordination and propagation mechanisms are expected to be implemented.

SMART has not been fully implemented and operated due to some business obstacles such as abrupt changes and cancellations of transactions. SEC is trying to stabilize its purchase orders within a fixed duration to minimize such changes and cancellations that drive its GMN into chaos. To accomplish the goals of SEC's GMN initiative, however, it is imperative that SMART play a key role to propagate transaction data with integrity. As suppliers in the lower tiers join the GMN, the amount of transaction data will be enormous. Without SMART supports, the KPI goals may not be satisfied smoothly.

The major success factors for a supply chain are effective management of strategic alliances, extensive data management capabilities, and advanced interorganizational information system to enable better information exchange (Liu et al., 2005). The mother company driven GMN has best potential in satisfying all of these success factors. The network may accomplish the best-in-class performance of a supply chain suggested in Levary (2000): (1) minimizing the bullwhip effect, (2) maximizing the efficiency of activities, (3) minimizing the inventories, (4) minimizing cycle times, and (5) achieving an acceptable level of inventory. The SEC case study proved the potential.

With the current state-of-the-art technology on hand, SMART was seamlessly implemented as Web services without degrading the whole system's performance. Without the sophistication functionalities of SMART, it is not easy to effectively control overall GMN operation and to accomplish the desirable goals intended. In that sense, SMART is recognized as a critical component of the GMN.

There are some limitations to SMART models. The current level of sophistication focuses on the transaction data propagation. Since a GMN tries to completely integrate entire suppliers to a virtual constellation of business processes, the sophistication level must be expanded from the transaction propagation to business intelligence propagation. Thus, SMART should be able to support top management in making strategic decisions regarding supply chain collaboration. Eventually, the scope of the integration will be expanded to enterprise portal (EP) level. Such EP-to-EP integration will include knowledge, workflow, key performance indicator (KPI), and mission and vision integration as well as data integration. Thus, SMART architecture will add gradually more sophistication functionalities as the integration level is expanded.

References

Andraski, J., "Leadership and the realization of supply chain collaboration", *Journal of Business Logistics*, Vol.19, No.2(1998), 9～11.

Barratt, M., "Understanding the meaning of collaboration in the supply chain", *Supply Chain Management*, Vol.9, No.1(2004), 30～42.

Bingi, P., M. K., Sharma, and J. K. Golda, "Critical issues affecting an ERP implementation", *Information Systems Management*, Vol.16, No.3(1999), 7～14.

Cowdrick, R. M., "Supply chain planning (SCP) -concepts and case studies", *Computers and Industrial Engineering*, Vol.29, No.1/4(1995), 271～281.

Dearden, J., "MIS is a Mirage", *Harvard Business Review*, Vol.50, No.1(1972), 90～99.

Frizelle, G. and E. Woodcock, "Measuring complexity as an aid to developing operational strategy", *International Journal of Operation and Production Management*, Vol.15, No. 5(1995), 26～39.

Frizelle, G., *The management of complexity in manufacturing*, Business Intelligence Limited, London, 1998.

Giachetti, R. E., "A Framework to review the information integration of the enterprise", *International Journal of Production Research*, Vol.42, No.6(2004), 1147～1166.

Gisolfi, D., "Web services architect, Part 1: An introduction to dynamic e-business", http://www-106.ibm.com/developerworks/library/ws-arc1/, 2001.

Green, F. B., "Managing the unmanageable: integrating the supply chain with new developments in software", *Supply Chain Management*, Vol.6, No.5(2001), 208～211.

Ireland, R. and R. Bruce, "CPFR: only the beginning of collaboration", *Supply Chain Man-*

agement Review, Vol.4, No.4(2000), 80～88.

Jacobson, I. et al., *Object Oriented Software Engineering,* Reading, MA, Addison-Wesley, 1992.

Kumar, K. and H. G. van Dissel, "Sustainable collaboration: managing conflict and cooperation in interorganization systems", *MIS Quarterly*, Vol.20, No.3(1996), 279～300.

Levary, R., "Better supply chains through information technology", *Industrial Management*, Vol.42, No.3(2000), 24～30.

Liu, J., S., Zhang, and J. Hu, "A case study of an inter-enterprise workflow-supported supply chain management system", *Information and Management*, Vol.42(2005), 441～454.

Malone, T., J. Yates, and R. Benjamin, "Electronic markets and electronic hierarchies", *Communications of the ACM*, Vol.30, No.6(1987), 484～497.

Mandry, T., G. Pernul, and A. W. Rohm, "Mobile agents in electronic markets: opportunities, risks, agent protection", *International Journal of Electronic Commerce*, Vol.5, No.2 (2000), 47～60.

McCarthy, S. and S. Golocic, "Implementing collaborative planning to improve supply chain performance", *International Journal of Physical Distribution and Logistics Management*, Vol.32, No.6(2002), 431～454.

Michael, W. and N. R. Jennings, "Intelligent agents: theory and practice", *The Knowledge Engineering Review*, Vol.10, No.2(1995), 115～122.

Moberg, C. P., T. W. Speh, and T. L. Freese, "SCM: Making the vision a reality", *Supply Chain Management Review*, Vol.7, No.5 (2003), 34～39.

O'Keeffe, M., "Building Intellectual Capital in the Supply Chain-the Role of E-Commerce", *Supply Chain Management*, Vol.6, No.4 (2001), 148～151.

Park, K. and A. Kusiak, "An ERP operations support system for maintaining process integration", *International Journal of Production Research*, Vol.43, No.9(2005), 3959～3982.

Poirier, C. C. and M. J. Bauer, *e-Supply Chain*, Berrett-Koehler Publishers, San Francisco, CA., 2000.

Premkumar, G. P., "Interorganization systems and supply chain management, An information processing perspective", *Information Systems Management*, Vol.17, No.3(2000), 56～69.

Ptak, C. A., *ERP: Tools, Techniques, and Applications for Integrating the Supply Chain*, St. Lucie Press, New York, NY., 2000.

Ross, D. F., *Competing through supply chain management: creating market winning strategies through supply chain partnerships*, Chapman and Hill, New York, NY., 1998.

Rothermel, K., F. Hohl, and N. Radouniklis, "Mobile agent systems: What is missing?" In Koenig, H., Geihs, K., and Preuss, T. (Eds.), *Proceedings of IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems*, Chapman and Hall, (1997), 111～124.

Savage, C. M., "Fifth Generation Management", *Dynamic Teaming, Virtual Enterprising and Knowledge Networking*, Butterworth-Heinemann, Newton, MA, 1996, 71.

Sabath, R. and J. Fontanella, "The unfulfilled promise of supply chain collaboration", *Supply Chain Management Review*, Vol.6, No. 4(2002), 24～29.

Shar, R., S. M. Goldstein, and P. T. Ward, "Aligning supply chain management characteristics and inter-organizational information system types: an exploratory study", *IEEE Transactions on Engineering Management*, Vol.49, No.3(2002), 282~292.

Sivadasan, S., J. Efstathiou, G. Frizzle, R. Shirazi, and A. Galinescu, "An information-theoretic methodology for measuring the operational complexity of supplier-customer systems", *International Journal of Operations and Production Management*, Vol.22, No.1(2002), 80~102.

Sproul, L. and S. Kiesler, *Connections: New ways of working in the networked organization*, MIT Press, Cambridge, MA, 1991.

Tidwell, D., *Web services-The Web's next revolution*, http://www6.software.ibm.com/developerworks/education/wsbasics/index.html, 2000.

van der Aalst, W., "Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries", *Information and Management*, Vol.37 (2000), 67~75.

Abstract

모기업 주도적 글로벌 생산 네트워크를 위한 조정 프레임웍

박광호*

본 논문은 모기업 주도적 글로벌 생산 네트워크에서 참여 기업간 교환되는 거래 데이터를 자율적으로 전파하고 조정하기 위한 프레임워크를 제시한다. 제시된 프레임워크는 공급사슬망의 성공적인 협업을 위해 발표된 선행 연구의 근본적인 개념에 기반을 두고 있으며 거래 데이터의 전파와 조정을 위한 모바일 에이전트를 적용하고 있다. 거대한 정보 웹에서 상호 연결된 거래 데이터의 무결성을 유지하는 것은 매우 어려운 과제이다. 프레임워크가 제시하는 조정 기능으로 글로벌 생산 네트워크를 효과적으로 통제할 수 있을 것이며 공급사슬망의 궁극적인 목표도 달성할 수 있을 것이다. 조정 기능의 현재 수준은 거래 데이터의 전파에 초점을 두고 있으며 향후 비즈니스 인텔리젼스 수준까지 확장할 계획이다.

Keywords : 공급사슬망관리, 협업IT화, ERP, 에이전트

저 자 소 개

박광호
현재 한양대학교 경상대학 경영학부 및 일반대학원 e-business경영학과 교수로 재직 중이다. 한양대학교 경영학과를 졸업하고 University of Iowa에서 경영학 석사 및 박사학위를 취득하였다. 삼성SDS에서 CIM컨설턴트, AI/UNIX팀장, 소프트웨어연구팀장 등을 역임하였으며, 주요 관심분야는 기업 정보화, ERP 교육, 정보시스템 개발 및 운영, 에이전트 시스템, 인공지능 응용 등이다.