FLEXIBLE AND ADAPTABLE PROCESS INFORMATION SHARING IN BUSINESS-TO-BUSINESS ELECTRONIC COMMERCE: FEDERATED DATABASE APPROACHES

Soon-Young Huh[1] and Kyoung-Il Bae[1]

Graduate School of Management[1]
Korea Advanced Institute of Science and Technology
P.O.Box 201 Cheongryang Seoul 130-012, Korea

ABSTRACT

This paper proposes the methodology through which independent business entities can share their process information with one another. The independent business entities have different business goals. Thus, in spite of their strategic alliance, they tend to be reluctant to share their core process information with one another. Besides, in agile organizations, their business relationships and internal business processes can be frequently changed to face the market environments. To meet such trends of modern business environments, in developing the methodology we especially reference the traditional federated database approaches and focus on two requirements for process information sharing, flexibility for representing the sharing policies established with respect to complex business relationships and adaptability for dealing with the dynamic changes of business relationships and internal business processes. Additionally, we suggest an object-oriented modeling to represent such sharing policies and shared process schemas.

KEYWORDS

Electronic Commerce, Information Sharing, Business Collaboration

1. Introduction

In the B-to-B electronic commerce, the business collaboration is mostly based on information sharing among collaborating business entities [5, 12]. Especially, sharing the inventory and process information of manufacturing, the business entities can make their systems more effective and accommodate their business goals such as just-in-time delivery, inventory cost minimization, and production bottleneck removal [4]. However, although the members are allied with others for their strategic purposes, they are still independent and voluntary. And, since their local systems have been developed independently to take best performance at domain specific operations, the systems are possibly heterogeneous. Because of these reasons, there are two obstacles in developing systems for information sharing. First, since the members have different business goals and opportunities, they tend to be reluctant to share their core business information with other collaborating members in spite of their strategic alliances [2]. Also, they want to operate their manufacturing systems independently. Second, changing business relationships (e.g., changing business partners) and changing internal business structures (e.g., updating business processes and inventory configuration) require additional development cost and time for information sharing [1, 7, 9].

For overcoming such difficulties and facilitating efficient business collaboration, this paper proposes a framework for flexible and adaptable information sharing. In developing the framework, traditional distributed database theories are adopted to effectively conceptualize the core constructs and mechanisms constituting the framework. The paper is organized as follows. In the chapter 2, we briefly review the federated database systems. Chapter 3 introduces our framework to facilitate flexible and adaptable process information sharing. Finally, chapter 4 summarizes this paper and proposes future research plans.

2. Federated Database Systems

For integrating distributed, autonomous, and heterogeneous local database systems, in 1985 Heimbigner and McLeod introduced federated database systems (FDBSs) [8] and in 1990 Sheth and Larson proposed a reference architecture which can be used as a methodology for developing FDBSs [11]. In this paper, we specifically reference the methodology proposed by Sheth and Larson. A FDBS is a collection of cooperating but autonomous component database systems (CDBSs) for which many database applications has been already implemented and dispersed widely in local sites [11]. The federated database system only allows partial and controlled sharing of data and the integrated database schema is composed of partial schemas of CDBSs. The reference architecture introduces five levels of database schemas including local schema, component schema, export schema, federated schema, and external schema. The local schema is the conceptual schema expressed with the native data model of the CDBS. Thus, most CDBSs have semantically and notationally different local schemas. To resolve such heterogeneity of local schemas, the federation should define a common data model that can represent the semantics of all local schemas. Using the common data model the local schema is translated into the component schema without semantic losses (transforming process). And, the component schema is narrowed to an export schema available to another participants of the federation (filtering process) and all participants' export schemas are integrated into one federated schema (constructing process). Then, participants make independently subsets of the federated schema, external schema, for their use (filtering process).

3. Process Information Sharing Framework

In developing the framework for process information sharing, we focus on defining following four steps.

STEP 1: Transforming Step

First step of the framework is to transform local information represented with a vender-specific format into semantically equivalent one with a canonical standard. Participants' local information is heterogeneous in the aspects of system platforms and implementation logics. Thus, for the purpose of communication, participants should determine a canonical standard for representing their business information. Industry standards such as Integration Definition for Function Modeling (IDEF) [10] and Workflow Management Coalition (WfMC) [14] can be chosen as the canonical standard. To perform this transforming step, mapping logics between vender-specific formats and the canonical standard are required. Usually, such mapping logics are encoded into transforming programs or stored in a persistent storage.

STEP 2: Filtering Step of Information Providers

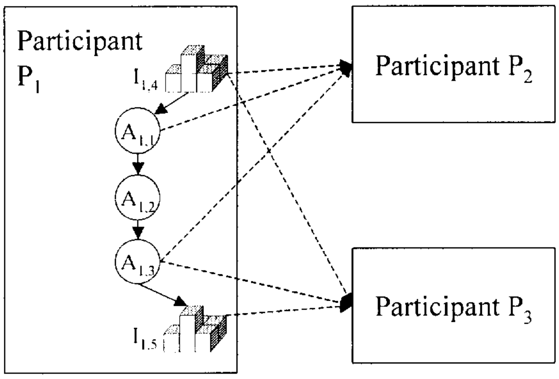
Second step of business federation model is to constrain the local information derived by step 1. Considering business relationships, each participant establishes sharing policies toward others and these policies are the keynotes in executing this step.

Note

- P_i: i-th participant of business collaboration
- P_i's j-th **activity supporter** $A_{i,j}$: a data set related to P_i's workflow activity
- P_i's k-th **inventory supporter** $I_{i,k}$: a data set related to one kind of material stored in P_i's inventory.
- P_i's n-th **supporter** $S_{i,n}$: a P_i's activity supporter or a P_i's inventory supporter, i.e., $\{S_{i,n}\} = \{A_{i,j}\} \cup \{I_{i,k}\}$

Sharing policies constitute of relationships describing which supporter can be shared to which participant and we define this relationship between a supporter and a participant as a **sharing dependency**. Figure 1 illustrates simple examples of sharing policies. Three participants, P_1, P_2, and P_3 are engaged in the example and P_1 has five supporters ($A_{1,1}$, $A_{1,2}$, $A_{1,3}$, $I_{1,4}$, and $I_{1,5}$). The dotted arrow from a supporter to a participant means a sharing dependency.

The concepts of supporter and sharing dependency help participants systematically establish their sharing policies to reflect business relationships with other participants. To perform this filtering step, supporters and sharing dependencies may be stored in a persistent storage. And, the sharing dependencies should be updated as corresponding sharing policies are changed.



Supporter	Participant
$A_{1,1}$	P_2
$A_{1,3}$	P_2
$A_{1,3}$	P_3
$I_{1,4}$	P_2
$I_{1,4}$	P_3
$I_{1,5}$	P_3

Figure 1. Sharing Policies and Their Corresponding Sharing Dependencies.

STEP 3: Constructing Step

While first and second steps of the framework are executed in information providing sides of participants, third step is done in information receiving ones. Third step is to integrate both internal supporters managed in local systems and external supporters received from external participants. In doing so, the constructing step involves two stages: constructing partial schemas by linking supporters and composing a full schema by linking the partial schemas.

<Construction of Partial Schemas by Linking Supporters>

We define a set of supporters provided by one participant and their associated links as a **partial schema**. To construct a partial schema, a participant first constructs a schema composed of all internal supporters. We call it an **internal schema**.

Note

- $SL(P_i, P_j)$: a set of supporters shared from P_i to P_j
- $S_{i,j} \rightarrow S_{i,k}$: there is a link from $S_{i,j}$ to $S_{i,k}$
- $S_{i,j} \Rightarrow S_{i,k}$: $S_{i,j} \rightarrow S_{i,k}$ or there exists a set of supporters $\{S_{i,n1}, S_{i,n2}, \ldots, S_{i,nm}\}$ such that $S_{i,j} \rightarrow S_{i,n1}$, $S_{i,n1} \rightarrow S_{i,n2}$, …, and $S_{i,nm} \rightarrow S_{i,k}$ (then we say that $S_{i,k}$ is **reachable** from $S_{i,j}$ and $\{S_{i,n1}, S_{i,n2}, \ldots, S_{i,nm}\}$ is a **path** from $S_{i,j}$ to $S_{i,k}$)

Using these definitions, Figure 2(a) writes up an algorithm for P_1 to create a partial schema of supporters shared to P_2 and Figure 2(b) exemplifies its process.

<Composition of a Full Schema by Linking Partial Schemas>

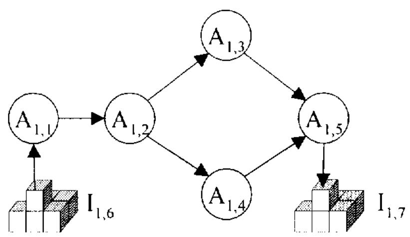
At this stage, we fully integrate the partial schemas constructed at the previous stage. We call the result of this stage a **full schema**. Participants may have supporters receiving raw materials or service orders from external participants, for example, inventories storing raw materials. We define this supporter as an **I-supporter**. On the other hand, participants may have supporters providing finished goods or services to external participants, for example, inventories storing finished goods. We define this supporter as an **O-supporter**. Our idea for composing a full schema is to make a link from an O-supporter of a partial schema to an I-supporter of another partial schema. We define such a link as an **external link**. This task needs the knowledge related to information and product flows between participants. Thus, while a participant uses only local business logics to construct its partial schemas, it should cooperate with other participants to compose a full schema.

The full schema helps participants specify the interrelationships with other participants and monitor whole activity and inventory information. And, it should be reconstructed as external participants' sharing policies are changed.

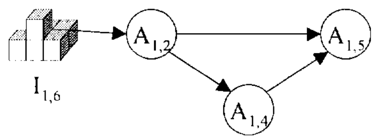
```
For each ordered pair (S_{1,i}, S_{1,j}) in SL(P_1, P_2) Loop
  If S_{1,i} => S_{1,j}
    For each path δ between S_{1,i} and S_{1,j}
      has_link = 1
      For each S_{1,k} other than S_{1,i} and S_{1,j} in SL(P_1, P_2) Loop
        If S_{1,k} is engaged in δ
          has_link = 0
        End If
      End Loop
      If has_link = 1 And IsExistLink(S_{1,i}, S_{1,j}) = 0
        MakeLink(S_{1,i}, S_{1,j})
      End If
    End Loop
  End If
End Loop

* If a link from S_{1,i} to S_{1,j} already exists,
  IsExistLink(S_{1,i}, S_{1,j}) returns 1, otherwise, 0.
* MakeLink(S_{1,i}, S_{1,j}) makes a link from S_{1,i} to S_{1,j}.
```

(a) Algorithm Constructing a Partial Schema

Internal Schema of P_1 for $SL(P_1)$'s All Internal Supporters



Partial Schema for $SL(P_1,P_2) = \{A_{1,2}, A_{1,4}, A_{1,5}, I_{1,6}\}$



(b) Examples

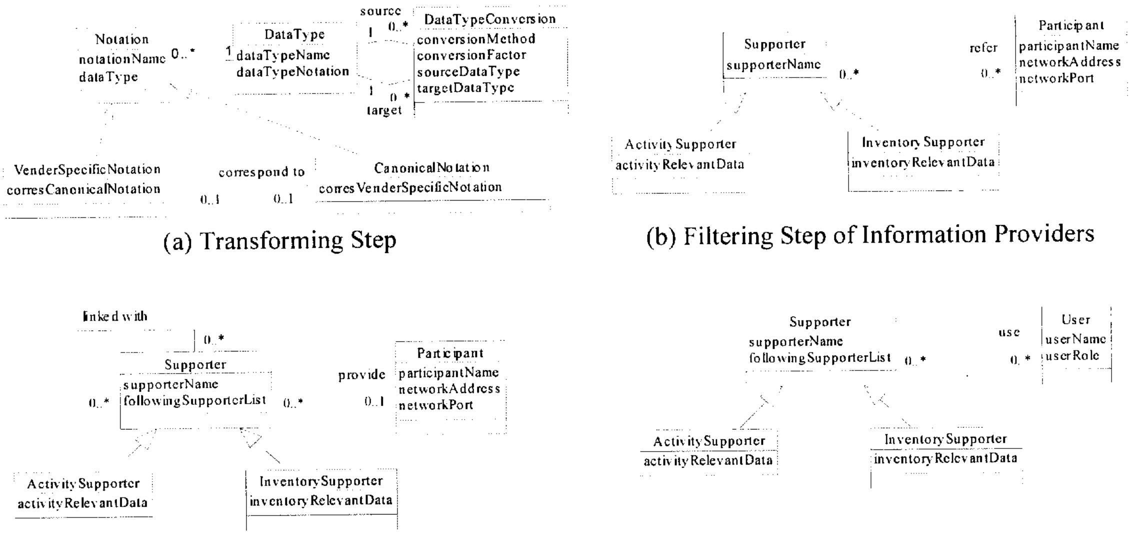
Figure 2. Constructing Partial Schema.

STEP 4: Filtering Step of Information Receivers

Fourth step chooses subsets of the full schema constructed by above steps. End users of participants do not refer to the full schema, but its subsets. We define this subset of a full schema as a **customized schema**. Reasons for the use of the customized schema are access control and customization. This step can be implemented by identifying which user can refer to which supporter. We define this relationship between a user and a supporter as a **security dependency**.

UML Class Diagrams Representing Schemas of Four Steps

In Figure 3, we illustrate an object-oriented modeling of the proposed framework by providing the class diagrams of the Unified Modeling Language (UML) [3].



(a) Transforming Step

(b) Filtering Step of Information Providers

(c) Constructing Step

(d) Filtering Step of Information Receivers

Figure 3. Class Diagrams of the Proposed Framework.

4. Conclusion and Research Plan

In B-to-B electronic commerce, efficiency of business collaboration depends on the information flows among business entities. In this study, we propose a framework for flexible and adaptable information sharing. The framework is composed of four steps including standardizing of business information (transforming step), establishing sharing policies (filtering step of information providers), constructing a full schema from internal and external supporters (constructing step), and customizing the full schema for internal use (filtering step of information receivers). Executing these steps, independent business entities can make different sharing policies corresponding to various business relationships. And, when the business entities confront the change of market environments the sharing policies can be easily updated without a significant amount of expertise or development cost.

With regard to this study, we are working for more rigorous formalism with mathematical tools such as graph theory and petri-net theory [6, 13] and are implementing prototype systems to show the applicability of the framework.

REFERENCES

1. Alonso, G., Fiedler, U., Hagen, C., Lazcano, A., Schuldt, H., and Weiler, N. (1999), WISE: Business to

Business E-Commerce, Proceeding of IEEE 9th International Workshop on Research Issues on Data Engineering, INFORMATION TECHNOLOGY FOR VIRTUAL ENTERPRISES (RIDE-VE `99), Sydney, Australia, March 23-24, 132-139

2. Bolcer, G.A. and Kaiser, G. (1999), SWAP: Leveraging the Web to Manage Workflow, IEEE Internet Computing, Jan.-Feb., 81-84
3. Booch, G., Rumbaugh, J., and Jacobson, I. (1999), The Unified Modeling Language User Guide, Addison Wesley, Massachusetts
4. Cachon, G.P. and Zipkin, P.H. (1999), Competitive and Cooperative Inventory Policies in a Two-Stage Supply Chain, Management Science, 45(7), 936-953
5. D'Amours, S., Montreuil, B., Lefrançois, P., and Soumis, F. (1999), Networked Manufacturing: The Impact of Information Sharing, International Journal of Production Economics, 58, 63-79
6. Ellis, C. and Nutt, G. (1993), Modeling and Enactment of Workflow Systems, Proceedings of 14th International Conference on Application and Theory of Petri-Nets, 21 Jun., Chicago, Illinois, 1-16
7. Hardwick, M., Spooner, D.L., Rando, T., and Morris, K.C. (1996), Sharing Manufacturing Information in Virtual Enterprises, Communications of the ACM, 39(2), 46-54
8. Heimbigner, D. and McLeod, D. (1985), A Federated Architecture for Information Management, ACM Transactions of Office Information Systems, 3(3), 253-278
9. Lee, H.B., Kim, J.W., and Park, S.J. (1999), KWM: Knowledge-based Workflow Model for Agile Organization, Journal of Intelligent Information Systems, 13(3), 261-278
10. National Institute of Standards and Technology (1993), Integration Definition for Function Modeling (IDEF0), Federal Information Processing Standards Publications 183
11. Sheth, A.P. and Larson, J.A. (1990), Federated Database Systems, for Managing Distributed, Heterogeneous, and Autonomous Databases, ACM Computing Surveys, 22(3), 183-236
12. Strader, T.J., Lin, F.R., and Shaw, M.J. (1998), Information Infrastructure for Electronic Virtual Organization Management, Decision Support Systems, 23, 75-94
13. Wilson, R.J. and Beineke, L.W. (1979), Applications of Graph Theory, Academic Press, New York
14. Workflow Management Coalition (1998), Interface 1: Process Definition Interchange Process Model