INTER-ORGANIZATIONAL PROCESS SIMULATION FOR PROCESS FOR INFORMATION SHARING DEVELOPMENT: TEST BED APPROACH

Kyoungil Bae　　Soonyoung Huh　　Junghyun Kim

Graduate School of Management
Korea Advanced Institute of Science and Technology
207-43, Cheongryangri-dong, Dongdaemun-gu, Seoul 130-012, Korea

ABSTRACT

Federated process framework is introduced responding to market's needs to gain benefits of process information sharing in a virtual enterprise. The framework is verified by implementing its prototype system. However, it is very difficult to integrate developed prototype system with operating legacy systems to show its applicability. Thus, a test bed approach is widely adopted to provide a system environment that imitates the operations of real legacy systems and their interactions. In this paper, we described implementation detail of test bed imitating the real-world workflow systems. We also show its contribution and results of the prototype system that was implemented on the test bed.

1. INTRODUCTION

In these days the Internet's World Wide Web has become the prime driver of contemporary electronic commerce. Recent studies by technology consulting groups predict more than one fourth of all business to business (B2B) purchases will be transacted on the Internet by 2004- a dollar volume 10 times that of Internet consumer purchases [6]. Like this, in the form of e-commerce, different companies join their services that can be executed over the Internet [1]. Besides, information sharing among collaborative organization produces lots of benefits. Process information sharing allows the removal of work redundancy that may exist in legacy system unnecessarily. By knowing exact information of sales and inventory, manufacturers can respond appropriately. They are able to reprioritize dealer orders, expedite shipments, and use overtime as well as plan production schedule. As a result, stock-outs are eliminated and lead-time is decreased. It creates great value for consumers in the form of lower price and greater availability of their favorite products. Resources saved from the new system's application can be reallocated properly. Also, by monitoring the status of activities, as the uncertainty of those activities become progressively lower, it is possible to continuously re-plan the overall process and optimize the result relative to simply waiting for each activity to complete [6]. From the views of customers, the reduction of lead-time or just in time delivery means the improvement of service. In addition, allowing customer to monitor the status of activity can be done in terms of service. Process information sharing means that each participant in a virtual enterprise provides internal process information to others while it utilizes external process information to perform local business operations more efficiently as the above mentioned. In spite of those many benefits, few system has developed yet to facilitate process information sharing in the Internet environment. Responding to these needs of market, a federated process framework was introduced. To verify the federated process framework introduced, implementation of prototype system is required. However, it is hard to integrate them with Real Operating Legacy Systems because of two main reasons. One is that the prototype system will affect to the legacy systems in either positive or negative ways and other is that there is no real system exactly matching to the prototype system. As a solution of these problems, test bed approach imitating the real-world workflow system is used commonly to prove if the prototype system is feasible and applicable or not

[8,2]. The test bed used here to verify the Federated process framework is developed with Visual Basic on the Windows NT operating system [10]. In this paper, we describe our experiences to develop a test bed to verify an academic system framework that is to facilitate process information sharing in an inter-organizational workflow. The paper is organized as follows: In the next section, we describe the federated process framework and prototype system in brief. In the third section, we describe the test bed development and in forth section, we show the result of the prototype system that was implemented on the test bed and contributions expected to existing inefficiencies of enterprises. In last section, we present the conclusion of this paper.

2. OVERVIEW OF FEDERATED PROCESS FRAMEWORK

The federated process framework enables facilitate process information sharing among interacting workflows systems in a virtual enterprise using intelligent agents and a Web-based process monitoring tool. We have used petri-net theory for logical validation of the framework and have developed a prototype system for physical validation. The Federated process framework is developed using the Object-oriented Database (OODB) [7] and Extensible Markup Language (XML). However, in the middle of developing the framework, we face two obstacles, the autonomy problem and the agility problem, that are inherent properties of modern organizations and make it difficult to develop effective process information sharing. To develop the federated process framework resolving those problems, we refer to a federated database system (FDBS) approach [3,12] and present general methodology that is composed of the following four steps. (1) local schemas of component database systems (CDBSs) are translated into other semantically identical schemas (component schema) using a standard data modeling method; (2) each component schema is restricted to a partial schema available to the FDBS (export schema) according to local data sharing policies; (3)multiple export schemas are integrated into one schema (federated schema); (4)finally, a federated schema is restricted to a schema that fits in with a task of a user or a user group (external schema).

3. TEST BED DEVELOPMENT FOR FEDERATED PROCESS FRAMEWORK

3.1 Overall Architecture

We introduce a prototype system of the federated process framework that includes object-oriented repositories, message-broadcasting agents, and a Web-based user view. The prototype system shows how the federated process framework can be implemented in the real system.
Figure 1 shows the overall architecture of test bed and the prototype system that has been developed based on the federated process framework [4].

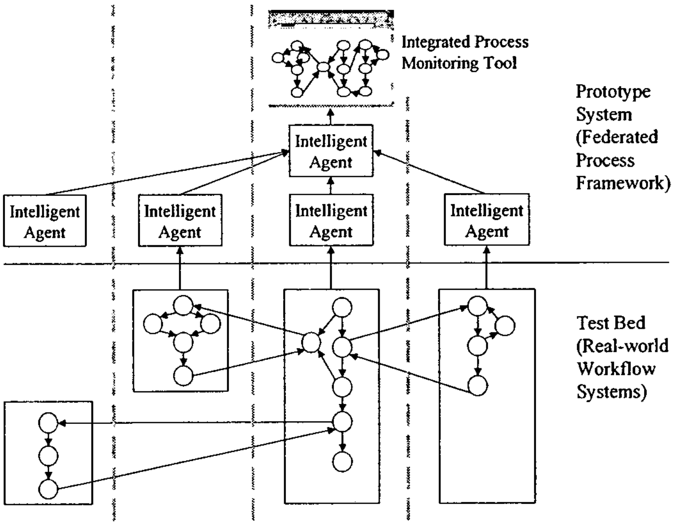


Figure 1. Overall Architecture of Test Bed and Prototype System

3.2 Assumptions

In developing the prototype system of the federated process framework, we take a new product development in food industry as a sample for inter-organizational process of product lifecycle management. Figure 2 illustrates the inter-organizational process in a virtual enterprise composed of a food company P1, a laboratory P2, a factory P3, and a design company P4. The local workflow of each participant Pi contains activities that comprise logical and independent pieces of work and we define Pi's j-th activity as Aij. In expressing various work sequences with work transition, the AND operator is distinguished from the OR operator [13] by providing the black dot in Figure 2. For example, when A1,3 terminates, it is followed by only one among A1,4 and A1,5 (OR-SPLIT).
On the other hand, when A1.5 terminates, both A1,6

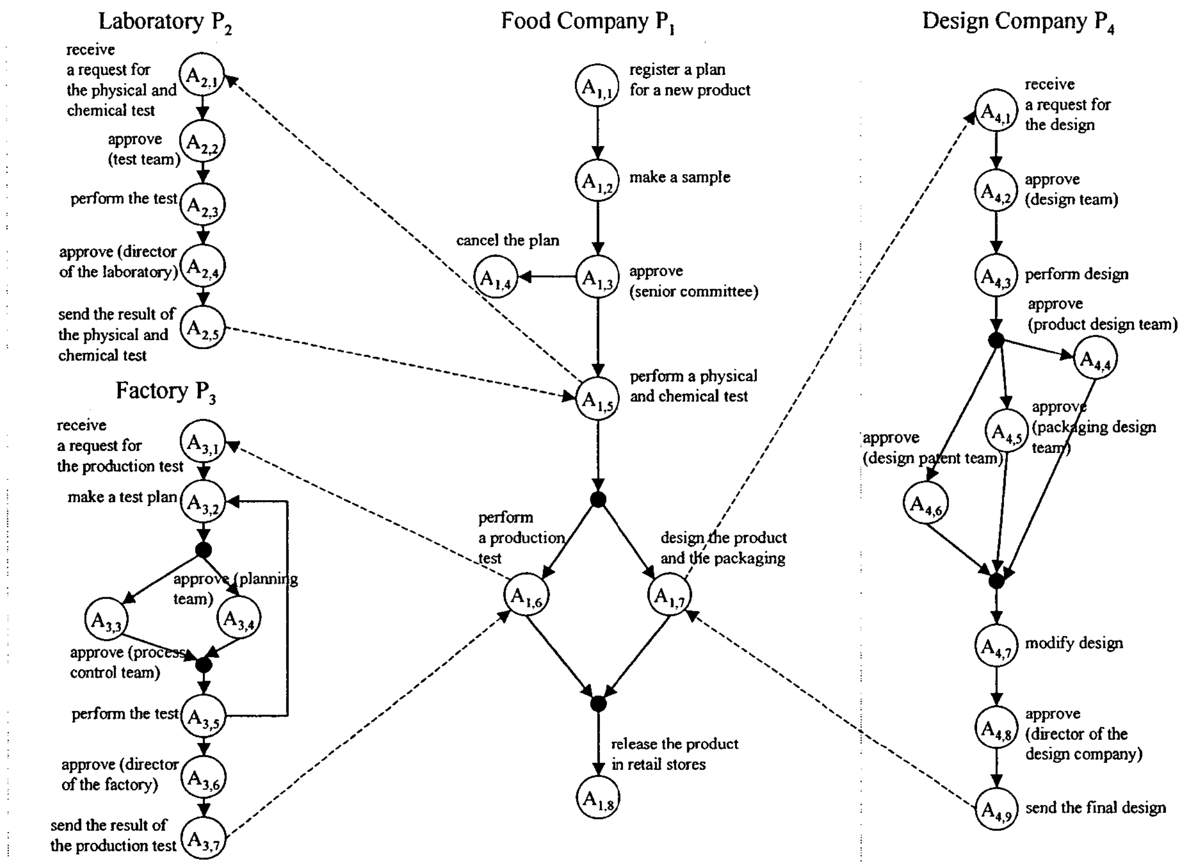


Figure2. Example of Inter-organizational Process in a Food Industry

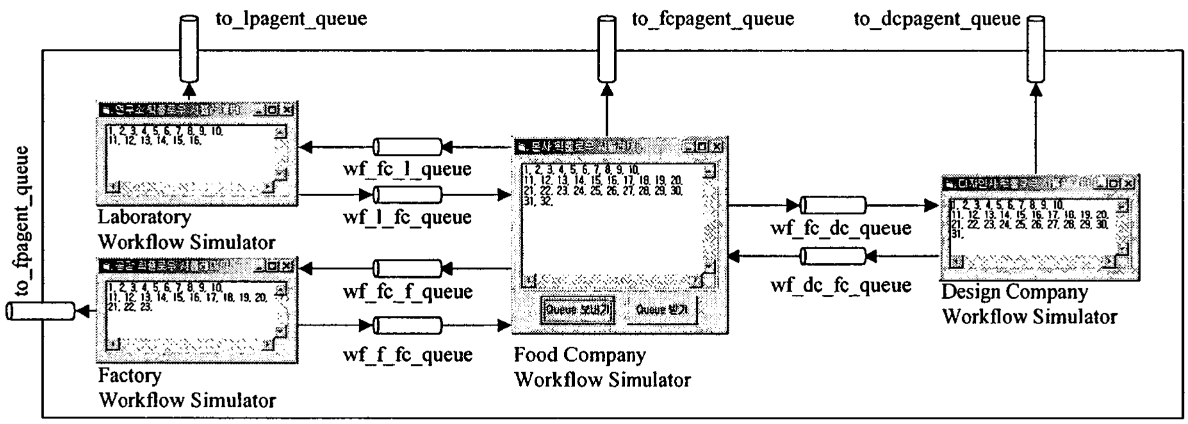


Figure3. Test Bed Implementation

and A1,7 (AND-SPLIT) follows it. In order to develop a test bed imitating this inter-organizational process, we need to specify the assumptions of the federated process system in the real -world workflow systems and then show its validity in the real-world workflow system. The assumptions that we specified are as follows.

-Every process instance at the run time in workflow systems proceeds according to the local pre-defined work sequences such as AND-JOIN, AND-SPLIT, OR-JOIN, and OR-SPLIT

-Every running activity in workflow systems should indicate one of three states, STARTING, RUNNING, COMPLETED.

-The interactions between workflow systems affect the execution of the workflow systems in a synchronous manner or in an asynchronous manner according to a collaborative relationship between the workflow systems.

-Each workflow system locally supports the subscribe-and-publish mechanism to notify the status of process instances to external applications.

The first three assumptions are followed by most software venders and standard organizations [9,13] as a common concept of workflow systems. Last one is optional functionality provided by several advanced workflow systems [6,10]. In case that a workflow system does not support the subscribe-and-publish mechanism, we can satisfy the forth assumptions through the implementation of a server process that periodically investigates the status of process instances from a workflow system and notifies it to external applications. Figure 4 visualizes the overall concepts of assumptions applied to development of test bed.

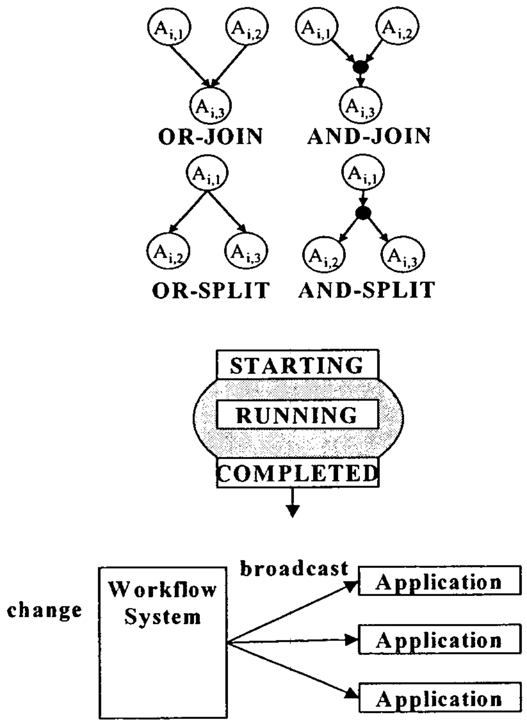


Figure 4. Visualized Concepts of Assumptions

3.3 Implementation of Test bed

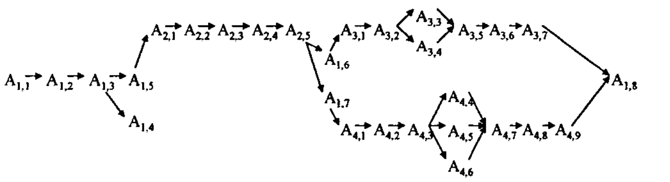
Figure 3 illustrates the test bed implementation in detail. The test bed consists of four workflow simulators and ten system queues. The former is used for representing the four local workflow systems as shown in Figure 2 and the latter for providing communication interfaces between separate software processes. Four workflow simulators are implemented to satisfy the assumptions presented in the previous section. Accordingly, they operate independently corresponding to pre-defined work sequences in Figure 2. System queues are adapted for the four workflow simulators to interact with one another at their execution time. And they play a role as system interfaces that are geared to notifying system messages (change notification messages) to external software processes when the status of internal workflow is changed. To implement system queues and to deliver outgoing message representing status of process instance, respectively, we used the Microsoft Message Queue (MSMQ) [10] and XML. Table 1 shows the description of ten queues in Figures 3.

Queue Identifier	Description
wf_fc_1_queue	messages from A1.5 to A2.1
wf_1_fc_queue	messages from A2.5 to A1.5
wf_fc_f_queue	messages from A1.6 to A3.1
wf_f_fc_queue	messages from A3.7 to A1.6
wf_fc_dc_queue	messages from A1.7 to A4.1
wf_dc_fc_queue	messages from A4.9 to A1.7
to_fcpagent_queue	messages to P1's agent
to_lpagent_queue	messages to P2's agent
to_fpagent_queue	messages to P3's agent
to_dcpagent_queue	messages to P4's agent

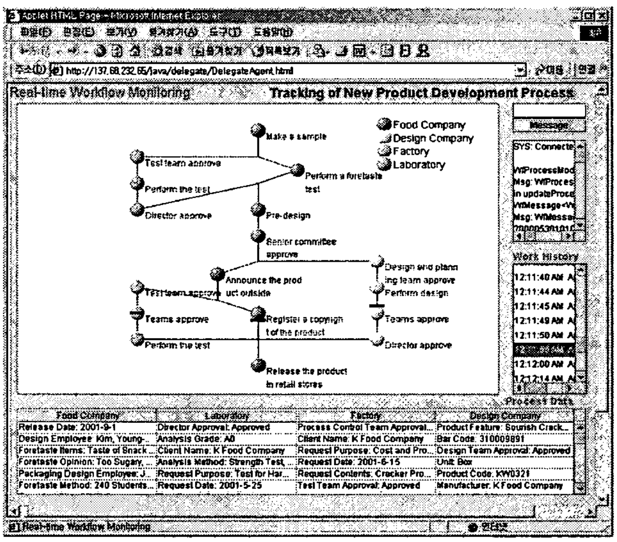
Table 1. Ten System Queues in Figure 3

3.4 Result of Prototype System Implemented on Test Bed

The test bed produces partially ordered change notification messages corresponding to the work sequences and the interactions among workflow systems in Figure 2. In Figure 5(a), we present the partial order of activities' initialization, which is described using arrows. Change notification messages in test bed follow this order. For example, A1,2 always precedes A1,3 and A2,3 but A3,1 and A4.1 do not start in any specific order. As shown in Figure 1, Test bed the change notification messages are delivered to the intelligent agents of the prototype system based on the federated process framework. Using the received messages, the integrated process monitoring tool provides a real-time process monitoring facility (see figure 5(b)) which is composed of three sections such as graphical process monitoring, process execution history, and process data monitoring sections. Accordingly, end users can track the work history of a process instance as well as the state of inter-organizational process. Thus, by implementing the prototype system on the test bed, we can show the feasibility and the applicability of the federated process framework.



(a) Partial Order of Activities' Starting Times



(b) Integrated Process Monitoring Tool of Prototype System

Figure 5. Results of Test Bed and Prototype System

4. CONTRIBUTIONS

The main contribution of this paper is that, by solving the autonomy problem and the agility problem, we provide a system framework and its detailed implementation methods for process information sharing that is adequate for the needs of modern business environments and current technologies. In particular, since the framework supports flexible sharing policies, it has wide applicability to various practical situations (e.g., loosely-coupled and tightly-coupled cases). Also, the proposed system architecture has strong adaptability to support the entire life cycle of process information sharing without serious maintenance cost and time. The OODB schema and the XML document structure mentioned briefly in second section contribute to the flexibility and the adaptability through detailed implementation methods that can be applied for actual development. Also, by virtue of the universality (i.e., vender-independency and platform-independency) of the XML, the XML document structure can be employed as a message specification among programs without requiring any modification at a development time.

As further technical contribution, first, there is scalability that the proposed system architecture provides in the case of new participant's entrance. When a new participant associates with existing process information sharing, little modifications are required through either adding the communication agents and repositories or revising their repository and associated PMs related to the new participant. Second, the proposed system architecture facilitates synchronous collaboration among participants in the Internet environment while providing strong software compatibility with existing Internet programs. Third, the proposed system architecture brings in various types of transparencies in managing sharing policies and broadcasting change notification messages.

5.CONCLUSIONS

In this paper, we introduce our experiences that we have obtained during development of a test bed imitating real world workflow systems. As assumptions considered in developing the test bed, we specify work sequences, activity states, interactions among workflow systems, and the subscribe-and-publish mechanism. These assumptions can be accepted in the current common concepts and technologies of workflow systems. According to the real inter-organizational process, the implemented test bed simultaneously executes workflow simulators representing independent workflow systems. The workflow simulators interact with one another at their execution time and sends change notification messages to the prototype system. Based on the messages received, the prototype system provides a real-time process monitoring facility to end-users. Consequently, using the test bed approach, we can show the feasibility and the applicability of the federated process framework through the prototype system can hardly be implemented on the operating legacy systems in the real world. Moreover, the introduction of system framework and its detailed implementation methods for process information sharing is main contribution made in this paper since it solved the autonomy and the agility problem that are crucial obstacles to develop process information sharing. In the future research, we are planning extension of the test bed to reflect the real delaying and running time of activities in executing workflow simulators and producing change notification messages. Then, we will be able to perform empirical tests in terms of dynamic production scheduling and logistics planning in an inter-enterprise process.

REFERENCES

[1] Alonso, G., Fiedler, U., Hagen, C., Lazcano, A., Schuldt, H., and Weiler,N. 1999. WISE: Business to Business E-Commerce. *IEEE.*

[2] David, N.V. et al. 1989. Development and Refinement of Text Bed Simulations. in *Proceedings of IEEE Energy Conversion Engineering Conference IECEC*-89: 315-320.

[3] Heimbigner, D. and McLeod, D. 1985. A Federated Architecture for Information Management. *ACM Transactions of Office Information Systems*, 3(3): 253-278.

[4] Huh, S.Y. and Bae, K.I, 2000. Flexible and Adaptable Process Information Sharing in Business-to-Business Electronic Commerce: Federated Database Approach. in *Proceedings of INFORMS-KORMS*, Seoul.

[5] IBM. 2000. *MQSeries Integrator*. http://www-4.ibm.com/software/ts/mqseries/integrator.

[6] Kuechler, W., Vaishnavi, V.K. Jr., and Kuechler, D. 2001. Supporting optimization of business-to-business e-commerce relationships. *Decision Support Systems 31.*

[7] Lamb, C. et al. 1991. The ObjectStore Database Systems. *Communications of ACM*, 34(10): 50-63.

[8] Law, A.M. and Kelton, W.D. 1991. *Simulation Modeling & Analysis*, McGraw-Hill.

[9] Leymann, F. and Roller, D. 2000. Production Workflow: *Concepts and Techniques*. Prentice Hall.

[10] Microsoft. 2001. http://www.microsoft.com/.

[11] Murata, T. 1989. Petri-nets: Properties, Analysis and Applications. in *Proceedings of IEEE* 77(4): 541-580.

[13] 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.

[14] Workflow Management Coalition (WfMC). 1996. Terminology & Glossary, Document Number WFMC-TC-1011.

AUTHOR BIOGRAPHIES

KYOUNGIL BAE a Ph.D. student at the Department Of Management Information Systems, KAIST. He received a B.A. degree in management science from KAIST in 1994, and M.S. degree from KAIST, Seoul in 1996. His research's includes Synchronous Collaboration in Groupware / Virtual Enterprise, System Integration in the Heterogeneous Computing Environments, Workflow and Process Automation. His e-mail address is kibae@kgsm.kaist.ac.kr

SOONYOUNG HUH an Assistant Professor of Management Information Systems at the Korea Advanced Institute of Science and Technology (KAIST). He received a B.S. degree in electronics engineering from Seoul National University in 1981, M.S. degree from the KAIST, Seoul in 1983, and a Ph.D. from the University of California, Los Angeles in 1992. Dr. Huh has published articles in such journals as Decision Sciences, International Journal of Intelligent Systems in Accounting, Finance and Management, Omega, Journal of Systems and Software. He has also presented his research at Conferences such as Decision Science Institute Conference and INFORMS. Dr. Huh's research deals with abstraction database application to business systems, model management, object-oriented database approach to decision support systems, and intelligent approaches in financial trading systems. His e-mail address and his web address are, respectively, syhuh@kgsm.kaist.ac.kr and http://kgsmweb.kaist.ac.kr/people/faculty/syhuh/.

JUNGHYUN KIM a research assistant of Applied Database Lab in Korea Advanced Institute of Science and Technology in South Korea. She received her BA in Spanish Language from Hankuk University of

Foreign Studies, Seoul, Korea in 1996 and MS in Information System from University of Colorado at Denver, Colorado, USA in 1999. Her research interests are in process model and data mining. Her e-mail address is jh0kim@kgsm.kaist.ac.kr.