한국시뮬레이션학회 2001춘계학술발표대회 논문집 2001.5.12 서울산업대학교

조직간 협동 프로세스 시뮬레이션을 위한 테스트베드 구축

배경일* · 허순영**

Test Bed Development for Inter-organizational Process Simulation

Kyoung-Il Bae, Soon-Young Huh

Key Words: Test Bed, Prototype System, Inter-organizational Process

Abstract

Advanced system architectures or frameworks proposed by researchers are usually verified by implementing their prototype systems, but it is very difficult to integrate them with operating legacy systems to show their 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 introduce our experiences to implement a test bed as a system environment on which we have implemented a prototype system to verify our studies for process information sharing. The test bed simulates simultaneously operating workflow systems and their interactions. We also show the result of the prototype system that was implemented on the test bed.

1. Introduction

Usually, it is very difficult to implement a prototype system interacting with operating legacy systems only for the purpose of research. Two main reasons for the difficulty are the influence on the legacy systems and the absence of real system environment adequate to the prototype system.

To overcome these problems, a test bed approach that imitates the operations of real-world systems is widely used to verify the feasibility and applicability of academic system frameworks [Law, 1991, David, 1989].

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 test bed was implemented using Visual Basic on the Windows NT operating system [Microsoft, 2000].

This paper is organized as follows: In section 2, we present the overall architecture of the test bed and the prototype system. Section 3 discusses the assumptions of the prototype system that was considered to implement the test bed. Section 4 describes the implementation details of the test bed and section 5 shows the result of the prototype system that was implemented on the test bed. In section 6, we present the conclusions of this paper.

2. Overall Architecture

In Figure 1, we illustrate the overall architecture of the prototype system that has been developed based on the federated process framework [Huh, 2000]. The federated process framework aims to facilitate process information sharing among interacting workflows systems in a virtual enterprise using intelligent agents

* 한국과학기술원 테크노경영대학원

** 한국과학기술원 테크노경영대학원

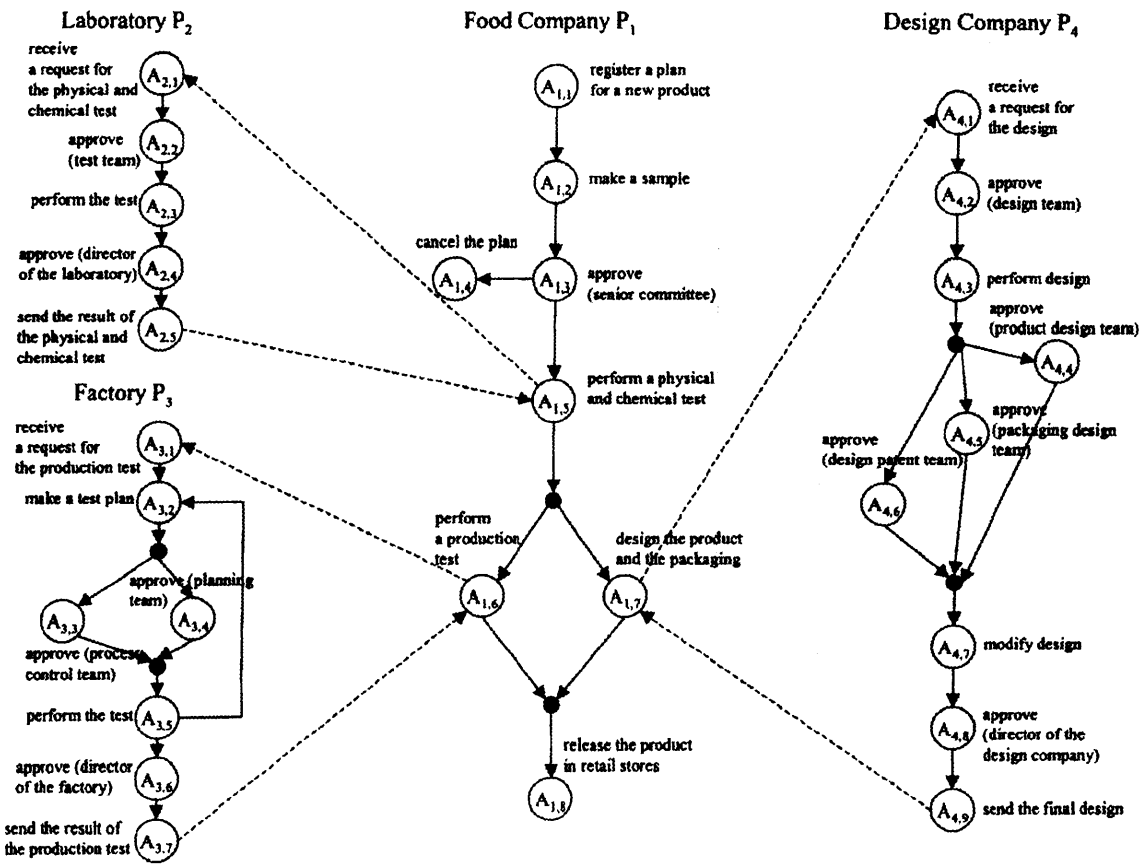


Figure 70. Example of Inter-organizational Process in a Food Industry.

and a Web-based process monitoring tool. In doing so, the framework provides both a scheme for process model integration and an object-oriented modeling for implementation in the Internet environment. We have first used petri-nets [Murata, 1989] to verify the framework analytically and have developed a prototype system on the test bed imitating the real-world workflow systems.

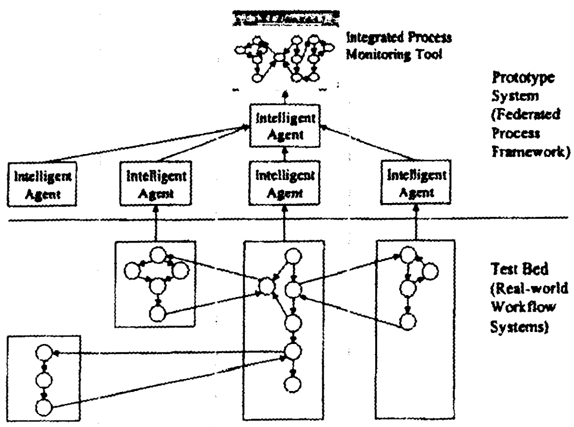


Figure 69. Overall Architecture of Test Bed and Prototype System.

3. Assumptions

In developing the prototype system of the federated process framework, we used an inter-organizational process of product lifecycle management to develop new products in a food industry. Figure 2 illustrates an example of the inter-organizational process in a virtual enterprise including 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 Ai,j. In expressing various work sequences with work transition, the AND operator is distinguished from the OR operator [WfMC, 1996] 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, it is followed by both of A1,6 and A1,7 (AND-SPLIT). To develop a test bed imitating this inter-organizational process significantly, we should first specify the assumptions of the federated process framework and show that they are valid in the real-world workflow systems. The assumptions of the federated process framework is specified 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 [WfMC, 1996].
- Every running activity in workflow systems should be one state among STARTING, RUNNING, COMPLETED.
- The interactions between workflow systems affect the execution of the workflow systems in a synchronous manner (e.g., P1 and P2 in Figure 2) or in an asynchronous manner (e.g., P1 and P3 in Figure 2) according to a collaborative relationship between the workflow systems.
- Each workflow system locally supports the subscribe-and-publish mechanism [Fenner 2000] to notify the status of process instances to external applications.

The first three assumptions are followed by most software venders and standard organizations [Leymann, 2000; WfMC, 1998] as a common concept of workflow systems. The final one is an optional functionality provided by several advanced workflow systems [IBM, 2000; Microsoft, 2001]. If a workflow system does not support the subscribe-and-publish mechanism, we can satisfy fourth assumption by implementing a server process that periodically investigates the status of process instances from a workflow system and notifies it to external applications.

4. Test Bed Implementation

Figure 3 illustrates the test bed implementation in detail. The test bed includes four workflow simulators representing the four local workflow systems in Figure 2, respectively, and ten system queues used for providing communication interfaces between separate software processes. To provide a test bed satisfying the assumptions presented in the previous section, we first implemented four distinct workflow simulators that operate independently corresponding to pre-defined work sequences in Figure 2. Using the system queues, they interact with one another at their execution time and also provide

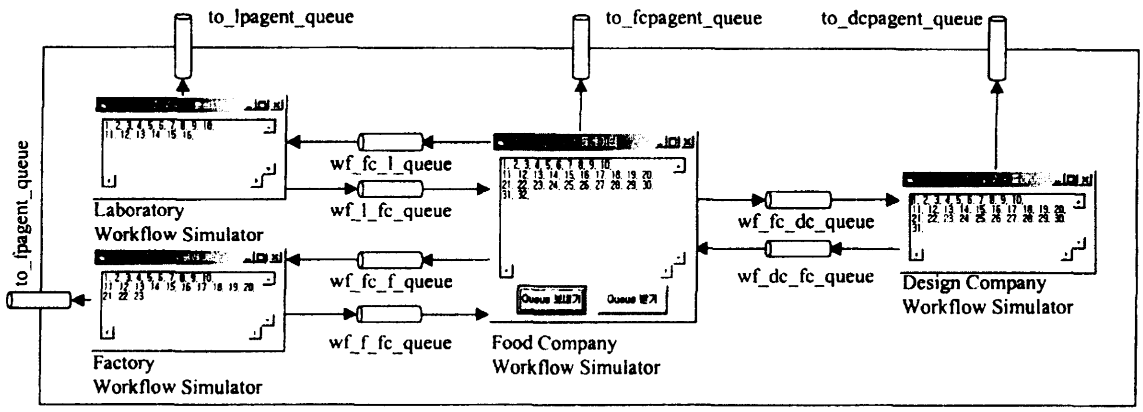


Figure 71. Test Bed Implementation.

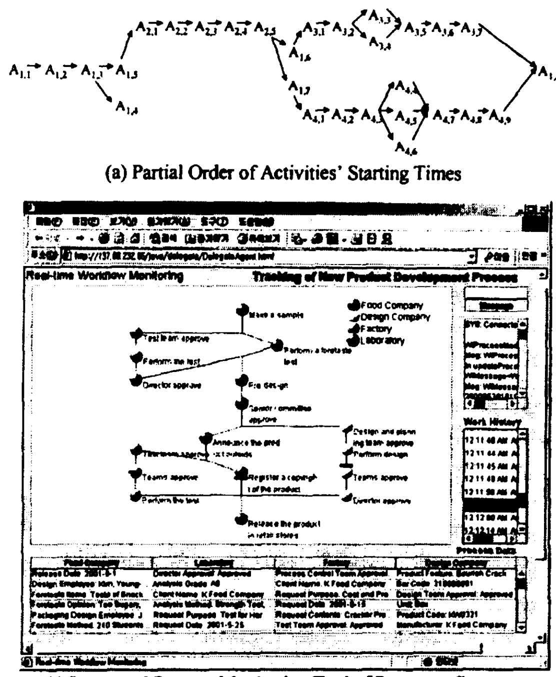
system interfaces that are geared to notifying system messages (change notification messages) to external software processes when the status of internal workflows is changed. To implement system queues, we used the Microsoft Message Queue (MSMQ) [Microsoft, 2001] and Table 1 shows the description of ten queues in Figure 3.

Queue Identifier	Description
wf_fc_l_queue	messages from A1,5 to A2,1
wf_l_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.

5. 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 Fiture 4(a), we present the partial order of activities' starting times (i.e., when activities become a STARTING state) and the implemented test bed produces change notification messages according to this order. The arrows of Figure 4(a) indicate the order of the starting time, 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, the intelligent agents of the prototype system based on the federated process framework receive change notification messages from the test bed. Using the received messages, the integrated process monitoring tool provides a real-time process monitoring facility (see Figure 4(b)) with which end users can tracks 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 72. Result of Test Bed and Prototype System.

6. Conclusions

In this paper, we introduce our experiences that we have obtained during development of a test bed imitating collaborating workflow systems in the real world. 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 of the federated process framework. From the messages, 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 though the prototype system can hardly be implemented on the operating legacy systems in the real world.

In future research, we are planning to extend 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 of the federated process framework in terms of dynamic production scheduling and logistics planning in an inter-enterprise process.

References

(1) David, N.V. et al., "Development and Refinement of Text Bed Simulations," in *Proceedings of IEEE Energy Conversion Engineering Conference IECEC-89*, 1989, pp.315-320.

(2) Fenner, B. et al., *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification*, Internet Engineering Task Force INTERNET-DRAFT draft-ietf-pim-sm-v2-new-01, 2000.

(3) Huh, S.Y. and Bae, K.I, "Flexible and Adaptable Process Information Sharing in Business-to-Business Electronic Commerce: Federated Database Approaches," in *Proceedings of INFORMS-KORMS*, Seoul, 2000.

(4) IBM, *MQSeries Integrator*, http://www-4.ibm.com/software/ts/mqseries/integrator/, 2000.

(5) Law, A.M. and Kelton, W.D., *Simulation Modeling & Analysis*, McGraw-Hill, 1991.

(6) Leymann, F. and Roller, D., *Production Workflow: Concepts and Techniques*, Prentice Hall, 2000.

(7) Microsoft, http://www.microsoft.com/, 2001.

(8) Murata, T., "Petri-nets: Properties, Analysis and Applications," *Proceedings of IEEE*, 77(4), 1989, pp.541-580.

(9) Workflow Management Coalition (WfMC), *Terminalogy & Glossary*, Document Number WFMC-TC-1011, 1996.

(10) Workflow Management Coalition (WfMC), *Interface 1: Process Definition Interchange Process Model*, Document Number WFMC-TC-1016-P, 1998.