e-Lollapalooza: A Process-Driven e-Business Service Integration System for e-Logistics Services

Kwanghoon Kim[1], *Member*, *KSII* and Ilkyeun Ra[2], *Member*, *KSII*
[1]Dept. of Computer Science, Kyonggi University,
San 94-6 Yiui-Dong, Youngtong-Gu, Gyeonggi-Do, 443-760, South Korea
[e-mail: kwang@kgu.ac.kr]
[2]Dept. of Computer Science and Engineering, University of Colorado at Denver,
Campus Box 109, P.O. Box 173364, Denver, Colorado, 80217-3364, USA
[e-mail: ikra@carbon.cudenver.edu]
*Corresponding author: Kwanghoon Kim

Received October 2, 2007; Revised November 22, 2007; Accepted December 1, 2007; Published December 25, 2007

Abstract

There are two newly emerging research issues in the enterprise information systems literature. One is the scalability issue for rapidly increasing choreographic volumes between interrelated organizations. The other is the business intelligence issue for traceable and monitorable business processes and services interchanging e-Business data and applications across organizations. Based upon these emerging issues, through a functional extension of the ebXML technology we have developed a process-driven e-Business service integration (BSI) system, which is named 'e-Lollapalooza'. It consists of three major components – the Choreography Modeler coping with the process-driven collaboration issue, the Runtime & Monitoring Client for coping with the business intelligence issue and the EJB-based BSI Engine coping with the scalability issue. This paper particularly focuses on the e-Lollapalooza's development aspects for supporting the ebXML-based choreography and orchestration among the engaged organizations in a process-driven multiparty collaboration for e-Logistics and e-Commerce services. Here, it is fully deployed in an EJB-based middleware computing environment for e-Logistics process automation and B2B choreography.

***Keywords*:** e-Business Service Integration System, B2B Choreography and Orchestration, e-Logistics

1. Introduction

There exist a lot of technologies which appear and disappear in our society. Among them, we can easily recognize that those technologies surviving until recently are those that have been helping to bring the world closer together either physically or conceptually. Furthermore, we can categorize those technologies into two groups according to the subjects of the world - people or organization - at which they are aiming. That is, one is a group of technologies targeted at bringing people closer together, the other is a group of technologies bringing organizations much closer together. The traditional groupware technologies such as realtime cooperative work solutions, intra-organizational workflows systems, enterprise resource planning (ERP) systems, electronic data interchange (EDI) solutions, and enterprise application integration (EAI) solutions are typical examples of the former, whereas the recent business process management (BPM) and web service (WS) technologies for e-logistics, e-commerce, e-market place, e-supply-chain management, and inter-organizational workflow systems are of the latter. Conceptually speaking, the former technology is focused on enhancing the efficiency and productivity of intra-organizational group activities. The latter was developed in order to promote inter-organizational collaborative activities. Consequently, we conclude that the focus of information technology be shifted from supporting the collaborative work of a group of people, towards supporting the collaborative work of a group of organizations.

In other words, the business process management and web services of the latter group and their related technologies have been swiftly adopted they are hot issues in the real world, and they are excellent topics for developing and discussing for the information technology market. This atmosphere is becoming a catalyst for triggering an explosion of the so-called process (business process) driven e-Business service integration domains, such as electronic logistics, electronic procurement, value-chain management, and supply chain management technologies. In addition, organizations are becoming much closer to each other and need to efficiently support their inter-organizational collaborative activities. We need to give special attention to ebXML in order to implement process-driven e-business service integrations. Traditionally, ebXML is a crucial technology and a standard for e-commerce. The vision associated with ebXML is to create a single global electronic marketplace where enterprise, without any concerns about size and geographical location, can meet and conduct their businesses with each other through the exchanges of XML-based messages. Thus, ebXML technology enables anyone to do business with anyone else in anyplace over Internet.

However, those process driven e-Business service integration models are slightly different from the traditional B2B model of e-Commerce, in terms of the type of behaviors among organizations in collaboration. That is, in the traditional B2B e-Commerce model, only dual-party (buyer/seller) organizations collaborate using the CPA/CPP of ebXML, which is established through agreement between the participating parties. However, it does not seamlessly provide complete solutions for trading procedures. This implies that we need to better secure ebXML against the illegal penetration of unauthorized people or organizations, rapidly increasing trade volumes, the very large variety of demands for customized services, the unchangeable requirements for traceable and monitorable business processes and services, etc. Without further modification, the ebXML technology should not be directly fitted into those process-centric applications of the process-driven e-Business service integration domains. Thus, in this paper, we try to extend the ebXML technology's functionality and, in particular, implement it in order to cope with the rapidly increasing trade volumes issue (the scalability issue) and the traceable and monitorable business processes and services issue

(the business intelligence issue). For the sake of the scalability issue, we adopt the approach of the EJB (enterprise java beans) framework [6][7] to implement a e-Business service integration system, and we implement original functionality providing traceable and monitorable business processes and services for the business intelligence issue. Based on these issues and considerations, we have completed the development of a process-driven e-Business service integration system, which is named 'e-Lollapalooza. In this paper, we describe the implementation details of the e-Lollapalooza system and its application, which is an outstanding example demonstrating how well the e-Lollapalooza system works in the e-Logistics and e-Commerce markets.

In the next section, we present the background and previously related works from the literature. In the subsequent sections, we describe the details of the e-Lollapalooza system such as the overall system architecture and components, process-driven e-Business collaboration model, choreography modeling tool, runtime & monitoring client, and the EJB-based e-Business service integration engine. Finally, we introduce an application example of the e-Lollapalooza system that collaborative organizations orchestrate for conducting the order and delivery process of a cyber-shopping mall.

2. Related Works

In South Korea, workflow/BPM and its related technological fields, such as B2B e-Commerce [1], ERP [2], SCM, CALM [3], EAI, etc., have begun to attract great attention from the community [2] of information science and database management, in respect to both aspects of research topics and also industrial issues such as information application fields. They are definitely capturing public attention. There are several projects involving ongoing research and development of the workflow and BPM systems [3][4][5][6][7][8] available from universities and research institutes, and even from industry. Of course, these technologies are also available and fairly well established in the worldwide information technology arena. In this paper, we are particularly interested in the B2B e-Commerce automation technologies. As a matter of fact, in order to accomplish total process automation for B2B e-Commerce, it is necessary for the workflow and BPM technology to be integrated, as a platform, with four major contributory technologies: object-orientation, EAI, web service, and XML technology. Thus, there might be two possible approaches, which we consider, to deploy a total process automation system for B2B e-Commerce – an open e-business framework (ebXML) [9][10][11][12] and a closed e-business framework (inter-organizational workflow and BPM [13]). Open e-business means that the business contractors or partners of the B2B e-business can be anyone who is registered in the registry/repository. In contrast to that, they are predetermined in the closed framework approach. We concluded that the former is a more reasonable approach for the process-driven e-Business service integration domain.

We would characterize ebXML as an open e-business framework because it is targeted at establishing a business contract between universal organizations unknown to each other. The ebXML is a set of specifications that enable the formation of a complete electronic business framework. If that the Internet is considered to be the information highway for e-businesses, then the ebXML can be thought of as providing the on-ramps, off-ramps, and the rules of the road [11]. It has been proposed by a joint initiative of the United Nations (UN/CEFACT) and OASIS, and has been developed with global participation for global use. This partnership brings a great deal of credibility to the notion that ebXML is representative of major vendors and users in the IT industry. It is supported by leading vertical and horizontal industry

groups including RosettaNet [14], OTA (Open Travel Alliance) and many others. Membership in ebXML is open to anyone, and the initiative enjoys broad industry support with hundreds of member companies, and more than 1,400 participants drawn from over 30 countries.

We will not describe the details of ebXML [12] here, but we adopt the basic concept and mechanism of ebXML as an e-business service agent for e-Lollapalooza, [15] which provides process-related information to e-business service entities across organizations. However, without modifications the ebXML is hardly applicable to the e-Logistics management or supply chain management framework, at which our e-Lollapalooza is targeted. This is because it is basically proposed for e-business contracts between two parties (buyer and seller), each of which corresponds to an organization in collaboration. That is, the concept of the business process being assumed in the ebXML domain is completely different from the concept in the e-Logistics domain. Thus, we try to extend its functionality so that it could be reasonably adopted for the process driven e-business service integration domain.

3. e-Lollapalooza: e-Business Service Integration System

In this section, we first define the basic concept of the process driven e-Business service integration (BSI) model which is slightly different from the traditional dual-party B2B model of e-Commerce in terms of the type of behaviors among organizations in collaboration. We also describe the functional details of the e-Lollapalooza system, such as its overall system architecture and its major components (EJB-based engine, e-business service integration process modeler and monitoring client). These are extended from the ebXML basic functionality and implemented in particular to cope with the rapidly increasing trade volumes issue and the traceable and monitorable business processes and services issue.

3.1 The Process-Driven e-Business Service Integration Model

The process-driven e-business service integration model [16] is about the specification of business contracts and the integration of the business services between organizations that are engaged in a process-driven multiparty collaboration. **Fig. 1** shows a simple example of a dual-party collaboration model consisting of three business transactions, and each business transaction performs two business activities - *Requesting activity* and *Responding activity* - that are conducted by the initiating role's organization and the responding role's, respectively.

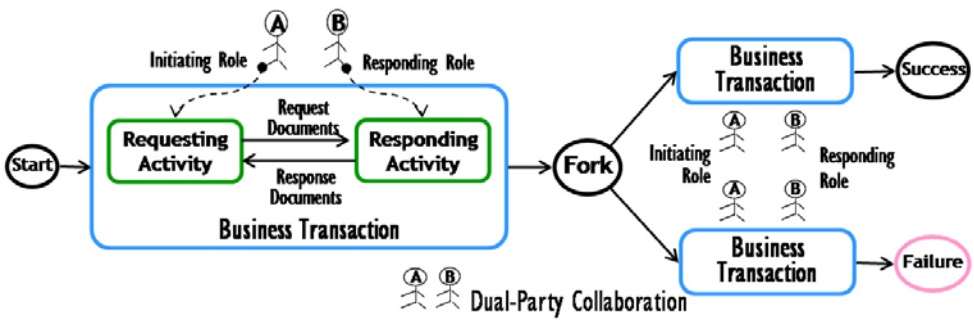

Fig. 1. A Binary Collaboration Service Model

This dual-party collaboration model becomes a binary collaboration model that represents a member of performs on a process-driven e-business service integration model as shown in **Fig. 2**, which shows a typical example of the process-driven e-business service integration model. The model consists of three dual-party collaborations (Binary Collaboration 1 ~ 3) that have certain types of control-precedence relationships, such as sequential, disjunctive or conjunctive, according to the business contracts of the organizations in multiparty collaboration.

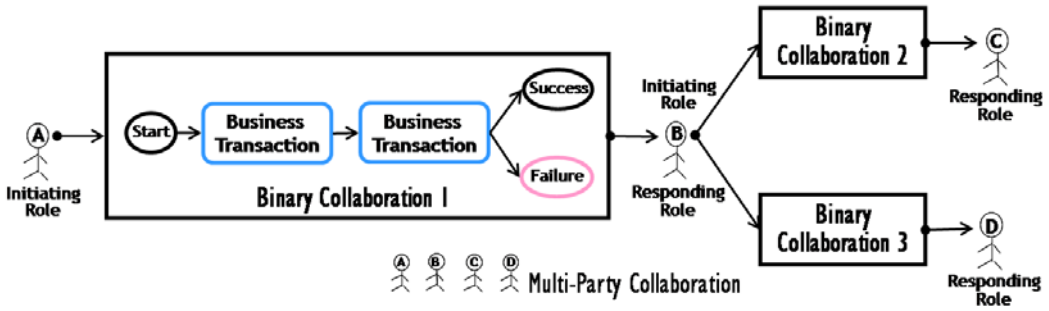

Fig. 2. A Process-driven Multiparty Collaboration Service Model

In addition, the dual-party collaboration is specified by the activity flow diagram, in which each activity represents a business transaction. Also, each organization (Organization A ~ D) is associated with either the initiating role, the responding role, or (for example organization B) both roles.

Additionally, each dual-party collaboration in the model is represented by the ebXML specifications. The ebXML's information models define reusable components that can be applied in a standard way within a business context. They enable users to define data that is meaningful to their business while also maintaining interoperability with other business applications. Also, the ebXML messaging service specification defines a set of services and protocols that enables electronic business applications to exchange data. The specification allows any level of application protocols including common protocols such as SMTP, HTTP, and FTP to be used. The Collaborative Partner Agreement defines the technical parameters of the Collaborative Partner Profiles (CPP) and Collaborative Partner Agreements (CPA). This captures critical information for communications between applications and business processes. It also records specific technical parameters for conducting electronic business.

Finally, the process-driven e-Business service integration model is fundamentally based on the open e-Business framework as we stated in the previous section. Thus, we need the Registry and Repository for the user applications in order to store company profiles and trading partner specifications. These mechanisms enable access to specific business processes and information models. They enable updates and additions over time. For the application developer it will store both the final business process definitions, and a library of core components. In particular, the ebXML Registry provides a set of services that enable sharing of information between interested parties for the purpose of business process integration between such parties based on the ebXML specifications. Such information is used to facilitate ebXML-based Business-to-Business (B2B) partnerships and transactions. As a result, a set of registry services for the process-driven e-Business service integration models is defined through the ebXML-based Registry Service Specification as shown in **Fig. 3**. It provides accessibility of registry contents to clients of the Registry.

3.2 Overall System Architecture of e-Lollapalooza

This section briefly concentrates on the architectural requirements and the components related to the design of the e-Lollapalooza business service integration system. The e-Lollapalooza has different runtime (enactment) architecture from conventional ebXML-based business service integration systems' in terms of process structure. We have conceived a particular conceptual architecture and functional implementation, which is suitable for the process-driven e-business service integration model and applications, such as electronic logistics, e-supply chain management, and process-driven e-commerce. Initially we present the architectural requirements of the e-Lollapalooza.

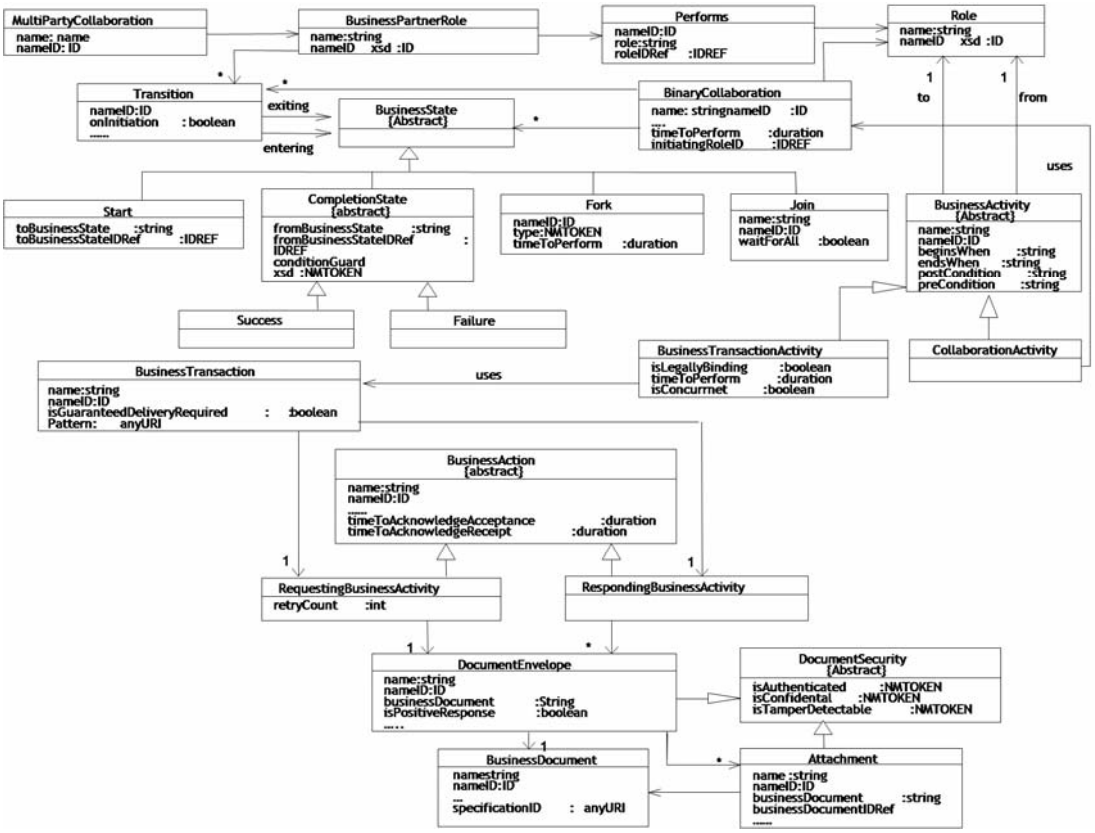

Fig. 3. The Schema of Process-driven e-Business Service Integration Model

The following requirements and design philosophies are applied to the e-Lollapalooza:

- Interoperability: This is the most important requirement particularly for the business service integration and choreography mechanisms to provide both cross-organization services and cross-platforms. Also, because various kinds of systems are deployed in a huge variety of domains, it is essential for the systems to be operational in heterogeneous computing environments. The e-Lollapalooza will be implemented on the EJB-based computing environment to preserve this interoperability requirement.
- Scalability: The e-Lollapalooza should concurrently provide consistent services to up to hundred thousands of business contract instances and several thousands of users. If the number of the instances increases, the performance should not decrease

by the same proportion. The ideal case is that the changes in performance of the e-Lollapalooza are linear under different workload situations.

- Extensibility: The e-Lollapalooza serves a huge variety of domains. Since the requirements within one domain over time, it is necessary for the system to be easily changeable and extendable in terms of its functionality according to the evolving demands and requirements. That is, the e-Lollapalooza is targeted at providing extendable and continuous functionality without disrupting any services.
- Distribution: Basically, the business contracts are executed in a distributed computing environment. Thus, the e-Lollapalooza should provide its functionality in a distributed fashion, too. Therefore, during its execution it must enable a workflow to be passed to a different machine, in order to redistribute the execution of the workflow. Also, a load balancing mechanism at the level of both workcases and workitems (activities) might be achievable in the e-Lollapalooza. This requirement can be safely satisfied by adopting the EJB framework [17] approach as an implementation platform. We recognize that these architectural requirements need highly advanced features and they are difficult to implement. However, they might be inevitable requirements in order for the e-Lollapalooza to be theoretically and technically acceptable for the markets targeted at electronic logistics and electronic commerce.

After considering the requirements, we developed the system that provides the basic functionality for defining, enacting and monitoring those process-driven e-Business service choreography and integration models. The overall system architecture and components are illustrated in **Fig. 4**. As shown in the figure, the e-Lollapalooza consists of three major architectural components - the BP modeler, the BP enactment engine, and the BP status monitor. These components asynchronously communicate with each other through the messaging system, and the models defined by the modeler are registered with the registry through the registry/registration component.

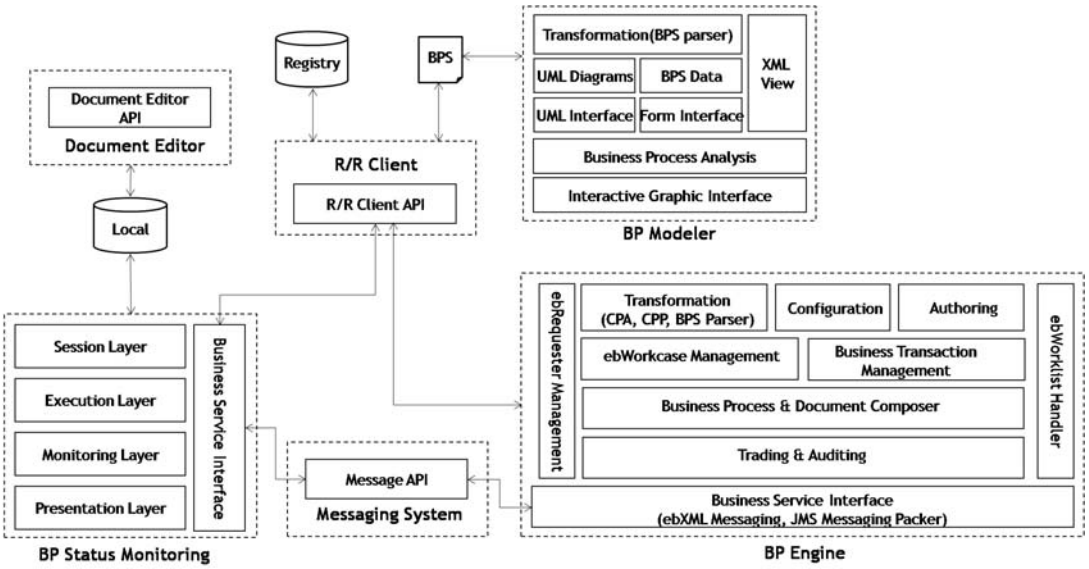

Fig. 4. Overall System Architecture and Components

The e-Lollapalooza system's design principle is summarized as followings:

- Basically the system is based upon the ebXML's framework for supporting a business collaboration and choreography between partner companies of the business integration services.
- Also, it is possible for the system to be deployed as a type of ASP (Application Service Provider) system in order to support business choreographs among small-sized business partners that are too small to be affordable to maintain a system.
- The conceptual architecture of the system is called the 'ebworkcase-based architecture' and it is originally proposed in this paper.
- The system is implemented by using the enterprise java beans (EJB) framework approach of Sun's J2EE [17][18].
- The communication mechanism of the system uses the EJB's JMS messaging technology and the ebXML's messaging technology.
- The BP status monitoring component of the system provides the runtime client functionality.

3.3 The e-Lollapalooza Business Process Modeler

The e-Lollapalloza BP modeler is a registry-based business service integration and choreography modeling tool which is tightly coupled with the registration and register client for process-driven e-business service integration models. Thus, the e-Lollapalooza BP modeler can be characterized by the concept of registry that is identical to ebXML's. In addition, it is implemented by the Java language and the EJB framework approach in order to be deployed on various platforms without any further technical consideration. Currently, the system is fully deployable and operable in an EJB-based computing environment.

Fig. 5 represents the components and their relationships (the class diagram) that comprise the modeler's package. The modeler package is composed mainly of four components - pkgbpmodeler.bps, pkgbpmodeler.model, pkgbpmodeler.utilexpression, and pkgbpmodeler.utilxml.

The component, pkgbpmodeler.bps, is for the specification (or attributes) of a business process that are needed for properly representing a process-driven e-business process integration model. It uses the component, pkgbpmodeler.utilexpression, to define the expressions needed in the model. Also, the attributes in the specification will finally be converted into the XML-based data format by the component, pkgbpmodeler.utilxml. The component, pkgbpmodeler.model, represents the graphical user interface of the modeler in order to provide a convenient way of defining a process-driven e-business process integration model. It also uses several utility components, such as pkgbpmodeler.model.awt.tree, pkgbpmodeler.model.dialog, pkgbpmodeler.model.event, pkgbpmodeler.model.graphics, pkgbpmodeler.model.util, and pkgbpmodeler.model.entity. **Fig. 6** is a set of the graphical icons that are used in the e-Lollapalooza modeler. **Fig. 7** and **Fig. 8** are captured screens of the models, created using the e-Lollapalooza modeler.

The former figure illustrates the use-case diagram and the activity diagram of a dual-party collaboration of the business service integration and choreography model. The latter shows a process-driven multi-party collaboration of the business service integration and choreography model. The window on the left-hand side of the screens represents the specifications of the e-business service integration models that are defined by a model designer. The right-hand sides of the screens are showing the ebXML-based specifications of the e-business service integration and choreography models. It can be seen that at the top of

the screen, there are several buttons. One of them is used to defining a multi-party (or process-driven) e-business service integration and choreography model. In particular, **Fig. 8** represents an example of the process-driven e-business service integration and choreography model proposed in this paper.

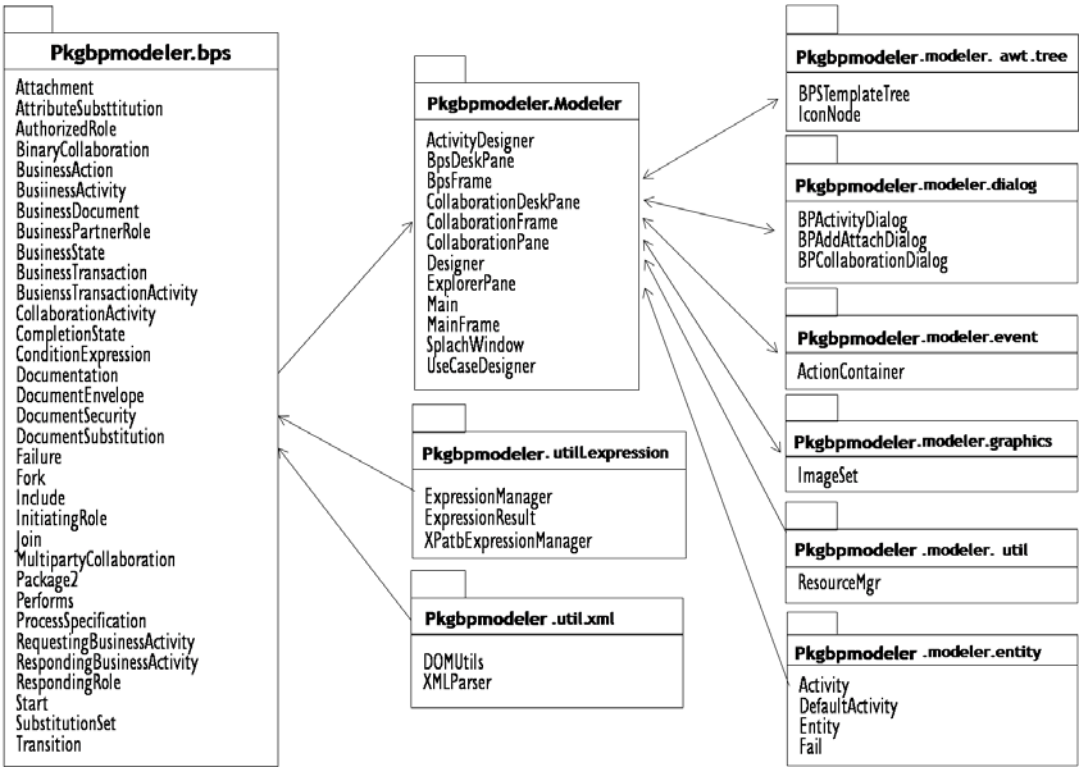

Fig. 5. The Class Diagram of the e-Lollapalooza Modeler

Icons	Meanings	Icons	Meanings
	Authorized Role/ Business Partner Role		Business Activity
			Fork
B T A	Business Transaction Activity		Join
			Transition
Performs	Performs		Success
	Start		Failure

Fig. 6. The Graphical Icons of the e-Lollapalooza Modeler

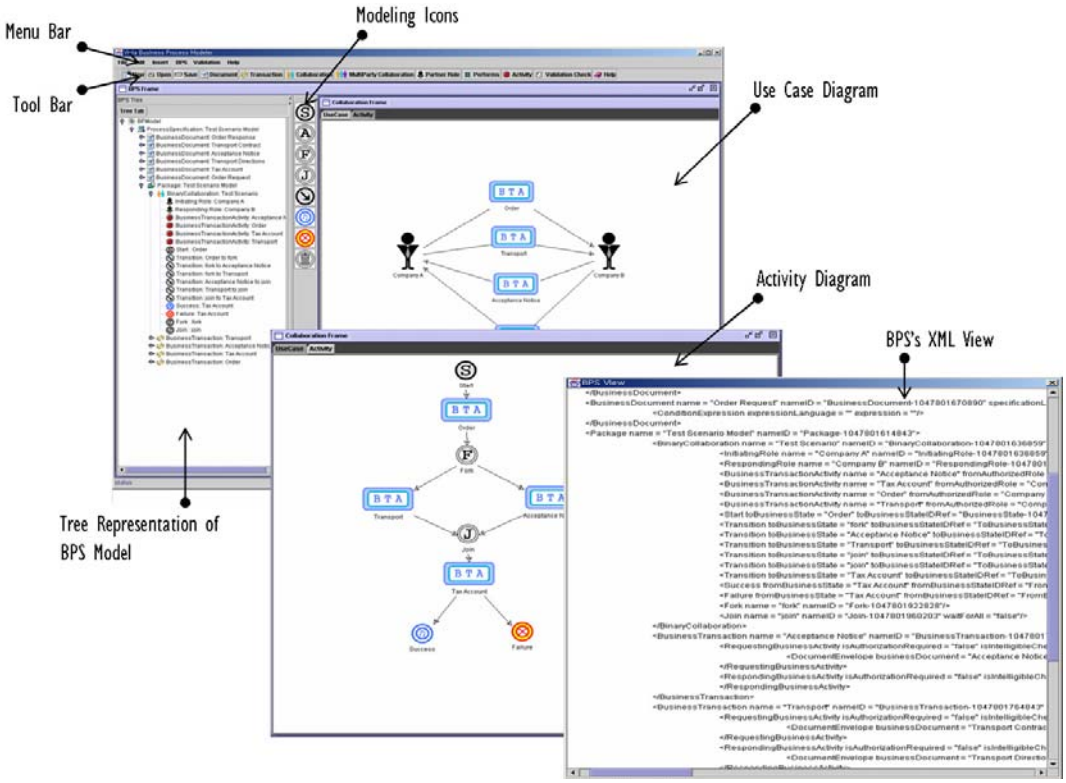

Fig. 7. An Example of the e-Lollapalooza Modeler: A Dual-Prarty Business Service Integration and Choreography Model

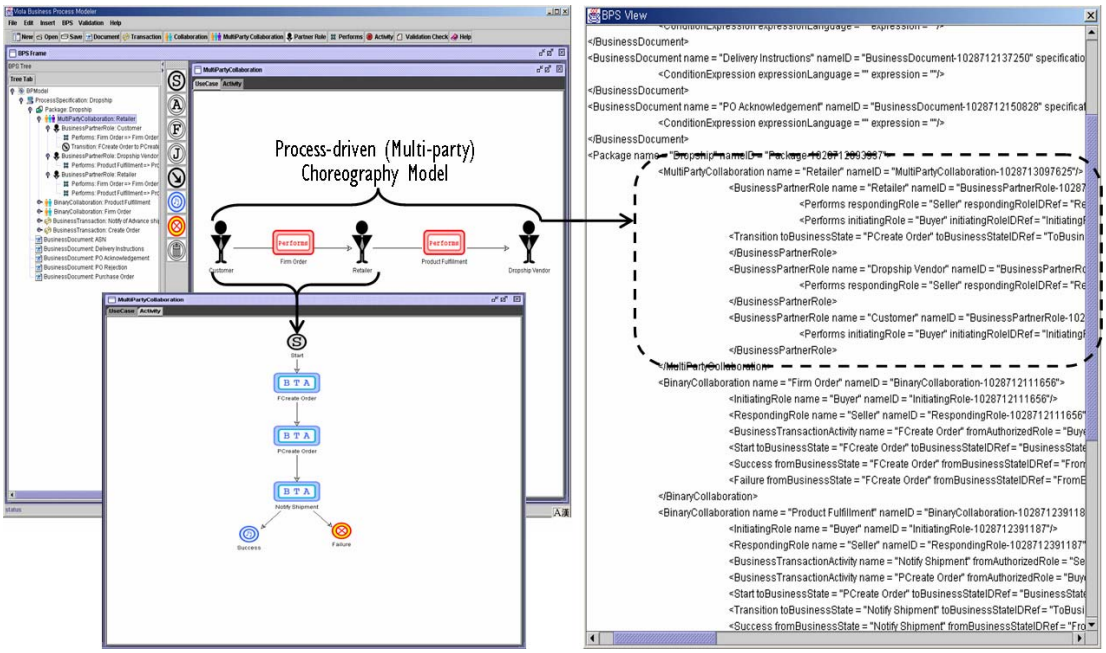

Fig. 8. An Example of the e-Lollapalooza Modeler: A Process-driven Business Service Integration Model

3.4 The e-Lollapalooza Business Process Enactment Engine

The functional architecture of the e-Lollapalooza's business process enactment part is illustrated in **Fig. 9**. It consists of the business process monitoring and the runtime client, the business service integration engine, and the message transfer system that supports the communication services between enactment engines by the use of secure (firewall protected) communication channels. All of these components are deployed in the EJB framework.

Thus, the communication between the monitoring client and the enactment engine is easily achieved by the use of the EJB simple object access protocol. However, the communication services between business service integration engines, each of which resides in each of the collaborative organizations, are performed by the use of the JMS-based message transfer system. Thus, a series of payloads for business contracts has to be wrapped by JMS message formats, and be passed to each of the underlining engines. The details of the monitoring client are explained in the next section. This section is dedicated to the description of the enactment engine.

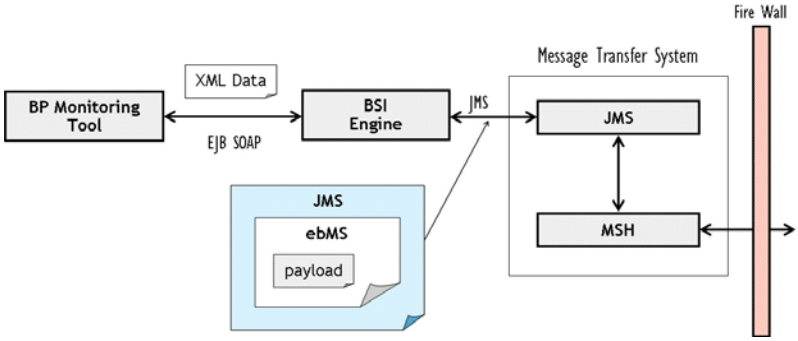

Fig. 9. The Architecture of the e-Lollapalooza BSIS's Engine

The internals of the business service integration engine are structured as shown in **Fig. 10**. In particular, in order for the e-Lollapalooza system to be satisfy the scalability requirement, we have proposed a new conceptual architecture that is called ‘ebWorkcase-based enactment architecture’. ‘ebWorkcase’ means the name of an object that is instantiated for dealing with its corresponding business service integration case in the real-world.

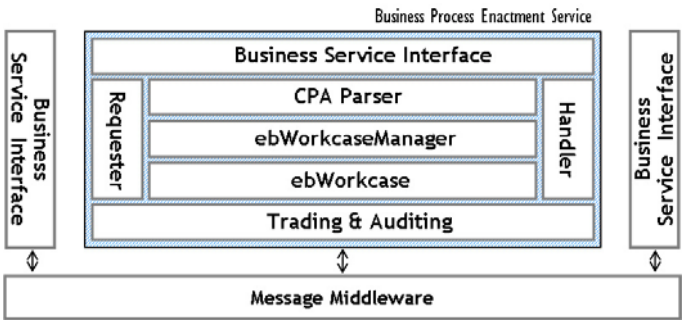

Fig. 10. The Functional Components of the e-Lollapalooza BSIS's Engine

Thus, it is reflecting the data structure of business choreography that contains a collaboration process, transaction activity and the execution order which comprises an ebXML-based and process-driven e-business service integration model. Also, it is based upon the CPA XML schema defined in ebCCP to support collaborations between partner companies by parsing transaction CPA, analyzing business process specification, and monitoring CPA execution status and progress reports. The ebRequester deals with a series of requests that informs ebWCM (ebXML Workcase Manager) creation or removal at the beginning of a transaction. The ebWCM manages the created ebWorkcase instances such as enacting instances, changing requested status information, and interacting with the monitoring clients. **Table 1** shows the detailed information and properties of the major functions performed by the ebWorkcase objects.

Table 1. Properties of the Major Functions handled by the BSIS's Engine

Business Collaboration	Business Transaction	Choreography within Collaboration
MultiPartyCollaboration: *name, Partners*	BusinessTransaction: *name, isGauranteedDeliveryRequired, precondition, postCondition, beginsWhen, endsWhen, pattern, activities, requester, responder*	BusinessState: *collaboration, entering, exiting*
BusinessPartnerRole: *performs, transactions, collaboration*	BusinessAction: *name, IsAuthorizationRequired, IsNonRepudiationRequired, isNonRepudiationOfReceiptRequired, tiemToAcknowledgeReceipt, isIntelligibleCheckRequired*	Transaction: *onInitiation, conditionGuard, conditionExpression, in, out*
Performs: *performedBy, authorizedRole*	RequestingBusinessActivity: *timeToAcknowledgeAcceptance, transaction, documentEnvelope*	Start
AuthorizedRole: *name, isInitiator, performers, from, to, collaboration*	RespondingBusinessActivity: *transaction, DocumentEnvelope*	CompletionState
BinaryCollaboration: *name, timeToPerform, precondition, postCondition, beginesWhen, endsWhen, pattern, role, states, usedBy, transitions*	DocumentFlow	Success: *conditionExpress*
BusinessActivity: *name, from, to*	DocumentSecurity: *IsAuthenticated, IsConfidential, isTamperProof*	Failure: *conditionExpress*
BusinessTransactionAcitivity: *timeToPerform, inConcurrent, isLegallyBinding, uses*	DocumentEnvelope: *isPositiveResponse, requesting, BusinessDocument, attachment*	Fork: *name*
BusinessCollaborationActivity: *uses*	BusinessDocument: *name, conditionExpression, documentEnvelope, attachment*	Join: *name, waitForAll*
	Attachment: *name, mimeType, specification, version, documentEnvelope, businessDocument*	

We have adopted an incremental (spiral) approach for the software development methodology in designing and implementing the e-Lollapalooza system. Thus, initially we

extract a set of usecases for the engine by considering the requirements as shown in **Fig. 11**. The usecases are the followings:

1. Business process management request usecase - It is in charge of the member and administrator roles by handling the requests for starting a collaboration process, and also receiving the administrator's process control requests and process start requests.
2. Process control usecase - It controls processes so that administrators are able to handle process creation, start, suspension, resumption, termination and completion.
3. Active business process monitoring usecase - It provides a set of services for the active business process's status inquiries.
4. Transaction documentation send-receive processing request usecase - It handles the requests from members to send-receive transaction documents through enactments of workitems in the worklist handler.
5. Business process documentation send-receive usecase - It deals with sending/receiving business process documents by making their relevant information messages.
6. Registry retrieve usecase - The registry stores CPAs, business process documentations, and business processes by the registration procedure.
7. To engine database save usecase - It enables the engine to store the statuses of processes' and the registry's contents.
8. Exception handling usecase - It deals with engine's malfunctions that occur during sending or receiving letters or transaction documents.
9. Audit data save usecase - It handles all traces of the engine's enactment history by keeping its audit logs.

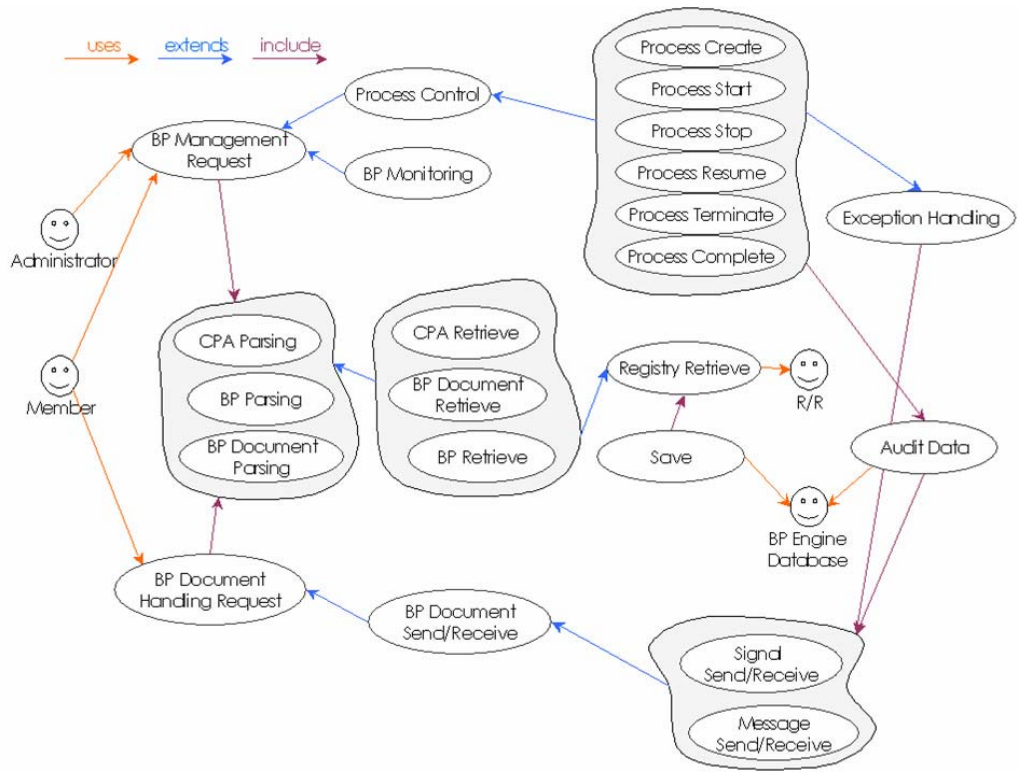

Fig. 11. The Usecases of the e-Lollapalooza BSIS's Engine

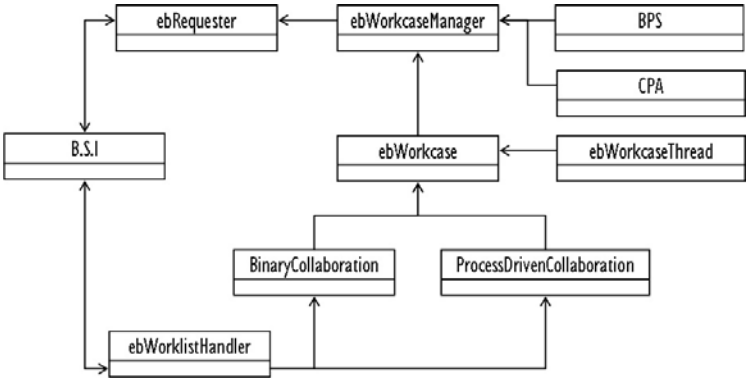

Fig. 12. The Class Diagram of the e-Lollapalooza BSIS's Engine

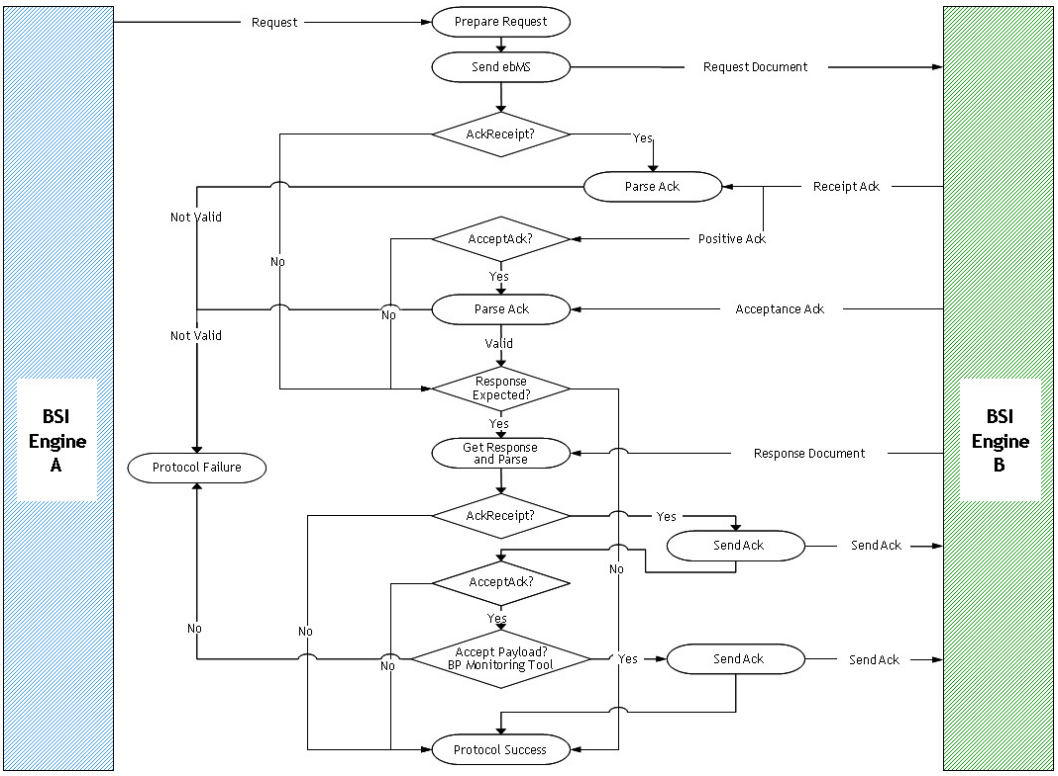

Fig. 13. The Transaction Managing Procedure of the e-Lollapalooza BSIS's Engine

Based on the usecases, we extract a set of classes and their relationships for the engine as shown in **Fig. 12**. The classes consist of BSI, ebRequester, ebWorklistHandler, ebWorkcaseManager, ebWorkcase, ebWorkcaseThread, BPS, CPA, BinaryCollaboration, and ProcessDrivenCollaboration. If ebRequester receives a business process instance creation and start request from the business process interface (BSI), then

ebWorkcaseManager parses its corresponding BPS and CPA by retrieving it from the registry. Also, by considering the business process's collaboration type (binary collaboration or multiparty collaboration), it creates its instance of ebWorkcase by the ebRequester. The ebRequester is basically inherited from the SessionBean of the EJB Framework. It delivers process creation and start requests to ebWorkcaseManager through the ebWorkcaseManagerHome. Finally, in **Fig. 13** we illustrate the transaction managing procedure performed by the collaborative engines deployed in the collaborating organizations.

3.5 The e-Lollapalooza Business Process Monitor

The business service monitoring tool is composed of two components - the monitoring server and the Business Service Integration Client. **Fig. 14** shows the CPA/BPS loading procedure of the BSI client. Also, the monitoring tool's operating examples are presented in **Fig. 15**, and **Fig. 16**. The former figure is a screen capture of the business transactions status window. The latter is of the activity flow status window. Each transaction's and each activity's status are distinguished by colors. In both the business transaction diagram and the activity diagram, grey, green and blue represent done, current-on-running, and tobe-on-running status, respectively. In this paper, we did not explain the design and implementation details of the business service monitoring tool.

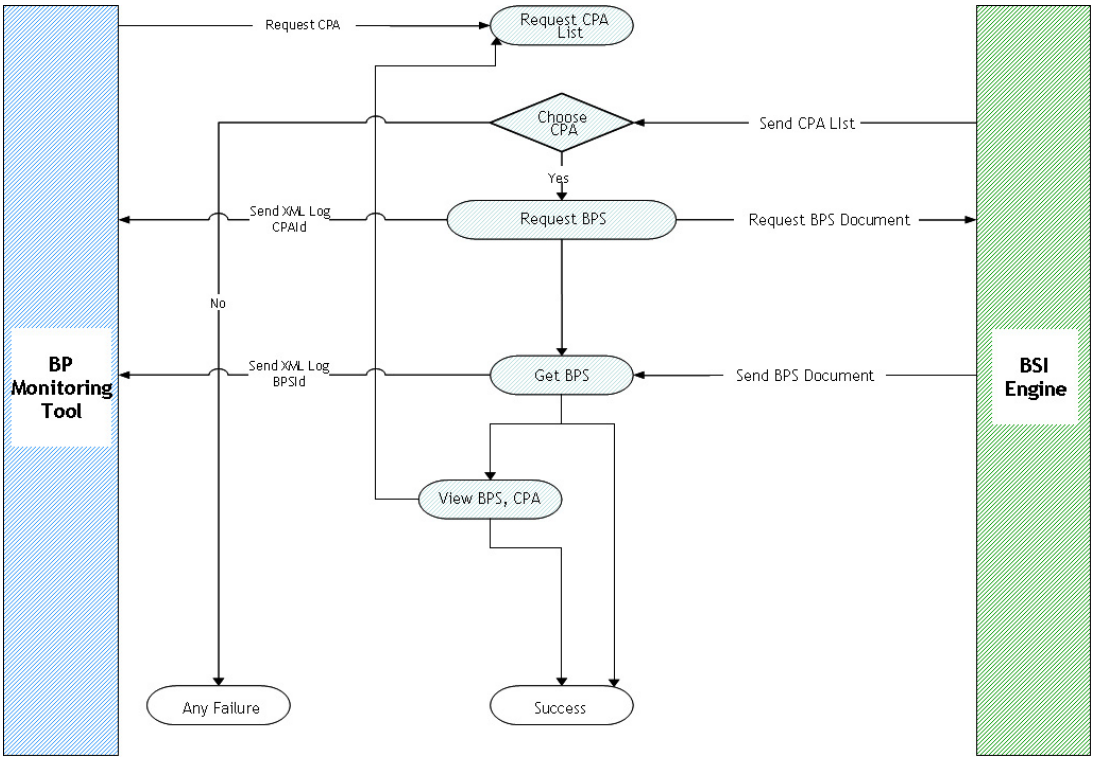

Fig. 14. The CPA/BPS Loading Procedure of the e-Lollapalooza BSIS's Monitoring and BSI Client

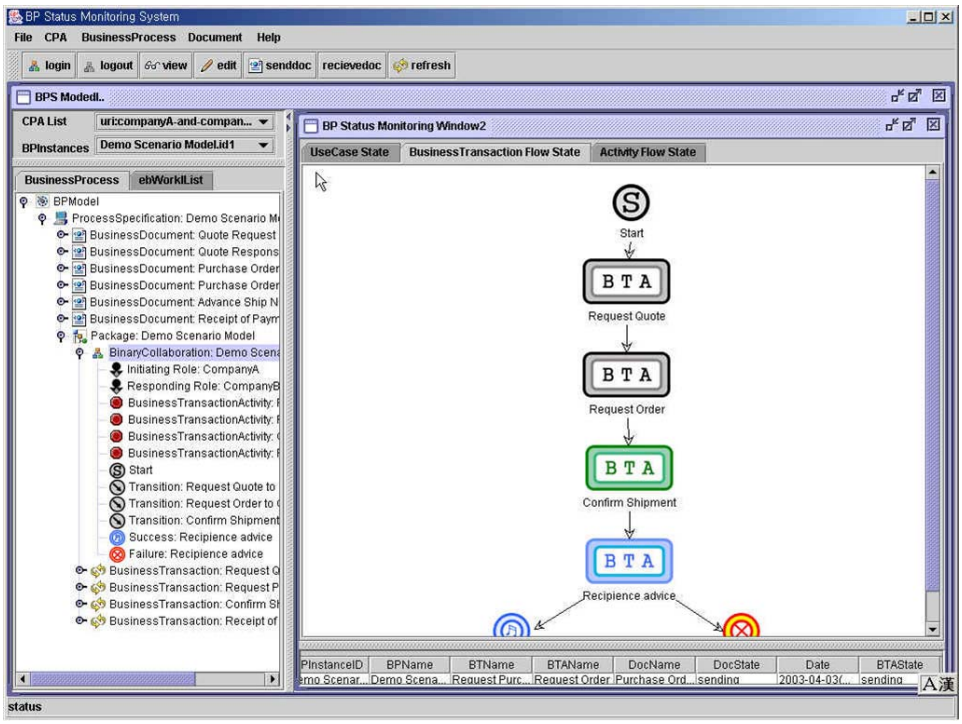

Fig. 15. The User Interface of the e-Lollapalooza BSIS's Monitor: Business Transaction Flow Status

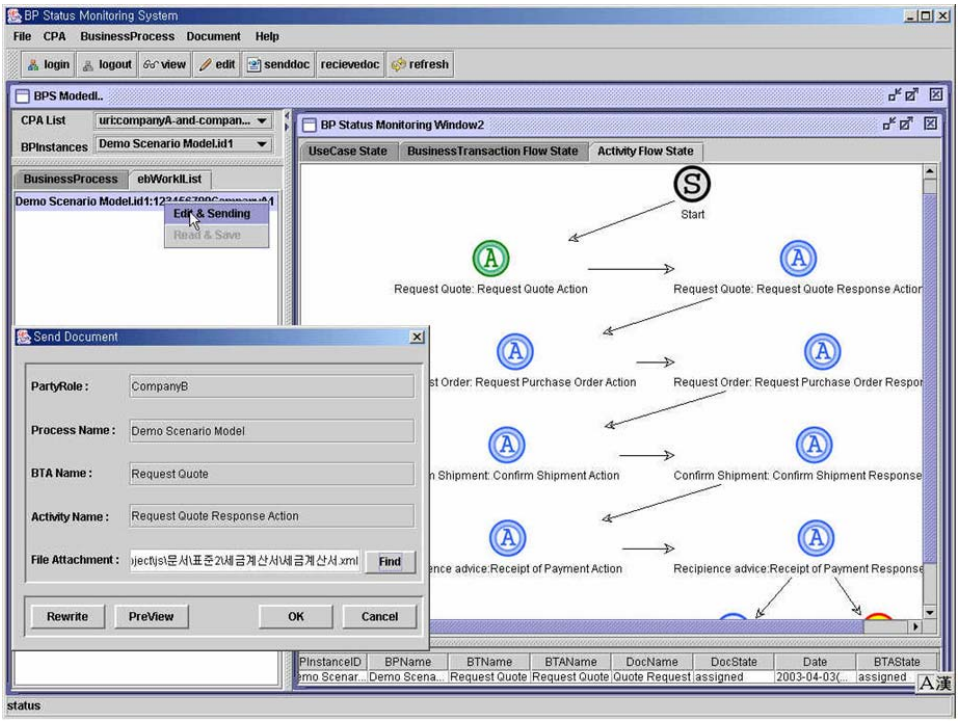

Fig. 16. The User Interface of the e-Lollapalooza BSIS's Monitor: Activity Flow Status

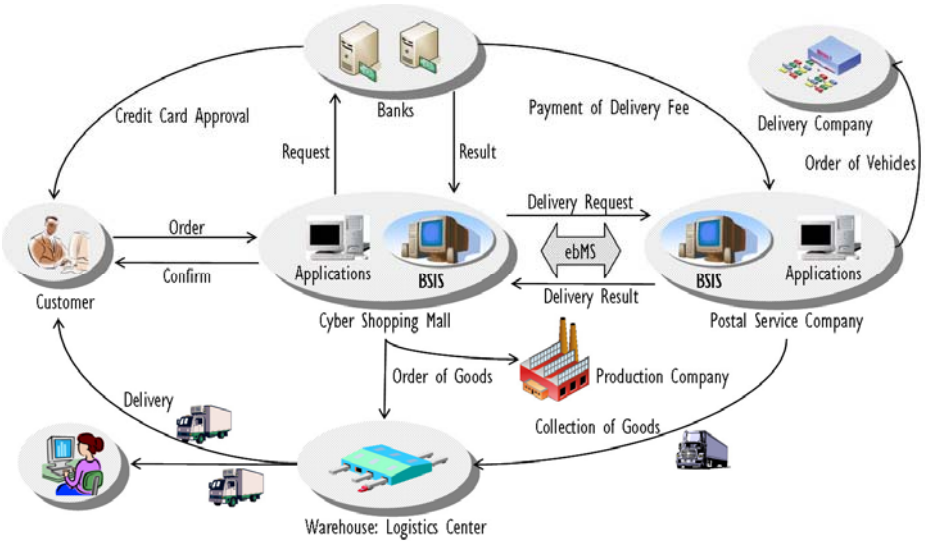

Fig. 17. The Application of the e-Lollapalooza BSIS: e-Logistic for the Cyber-Shopping-mall of a Postal Company

3.6 An Application of the e-Lollapalooza: e-Logistics for the Postal Cyber-Shopping Mall

Fig. 17 shows an e-Logistics application of a process-driven inter-organizational business service integration model to which the e-Lollapalooza BSI system has been applied. The process-driven inter-organizational choreography processes between the cyber-shopping mall and the postal service company are modeled by the e-Lollapalooza choreography modeling system. They are integrated and serviced by the BSI engine systems that are installed in both of these two companies. Currently, all the application programs of the e-Logistics example including contract and order document forms are fully available on the web site, http://www.cnit.com.

4. Conclusions

To this point, we have described the implementation details of the e-Lollapalooza process-driven inter-organizational business service integration system. It is targeted at the process-driven e-business service integration markets, such as e-Logistics, e-SCM, e-Procurement, and e-Government. These require process-driven multi-party collaborations by a set of independent organizations. The process-driven choreography models specifying the dual-party collaboration and the process-driven multi-party collaboration between organizations are transformed into the ebXML-based specifications. Finally, the e-Lollapalooza e-business service integration (BSI) system is able to enact and control the choreography models.

Both In South Korea and in world-wide research arena, process-driven e-business services and their related technological fields, such as e-SCM, e-Procurement, e-Logistics, e-Government, are receiving great attention from industry in the information management and business process management fields. Thus, there are many challenges to develop and commercialize e-business solutions. This paper is one of the active attempts at pioneering process-driven inter-organizational choreography systems which support cross

organizational e-business processes and service integrations. In further research works, we will apply the e-Lollapalooza system to various fields of the cross-organizational e-business processes and service integrations markets.

Acknowledgement

The Research was supported by the Contents Convergence Software Research Center, 2007-081-0, funded by the GRRC Program of Gyeonggi Province, South Korea.

References

[1] Dogac, A., Tambag, Y., Pembecioglu, P., Pektas, S., Laleci, G. B., Kurt, G., Toprak, S., Kabak, Y., "An ebXML Infrastructure Implementation through UDDI Registries and RosettaNet PIPs", *ACM SIGMOD International Conference on Management of Data*, Madison, Wisconsin, USA, June 2002.

[2] H. Fosdick, "The Sociology of Technology Adaptation", *Enterprise System Journal*, September 1992.

[3] Haruo Hayami, Masashi Katsumata, Ken-ichi Okada, "Interworkflow: A Challenge for Business-to-Business Electronic Commerce", *Workflow Handbook 2001*, *WfMC*, October 2000.

[4] Kwang-Hoon Kim, Clarence A. Ellis, "A Framework for Workflow Architectures", University of Colorado, Department of Computer Science, Technical Reports, CU-CS-847-97, December 1997.

[5] Dong-Keun Oh, Kwang-Hoon Kim, "An EJB-Based Database Agent for Workflow Definition", *Journal of Korean Society for Internet Information*, *Vol.4*, *No.5*, December 2001.

[6] Kwang-Hoon Kim, Clarence A. Ellis, "Performance Analytic Models and Analyses for Workflow Architectures", *Information Systems Frontiers*, *Vol. 3*, *No. 3*, pp. 339-355, 2001.

[7] Sung-Su Sim, Kwang-Hoon Kim, "Workcase based Very Large Scale Workflow System Architecture", *Proceedings of Korea Database Society*, October 2002.

[8] Dong-Keun Oh, Jong-Mi Chun, Jung-Sun Hong, Se-Won Oh, Jae-Gak Hwang, Yong-Jun Lee, Kwang-Hoon Kim, "ebXML-Based e-Logistics Workflow Engine Architecture", *Proceedings of the Conference of Korean Society for Internet Information*, *Vol 3*, *No.3*, pp. 348-351, November 2002.

[9] UN/CEFACT and OASIS - ebXML Registry Information Model v1.0, May 2001.

[10] UN/CEFACT and OASIS - ebXML Registry Services Specification v1.0, May 2001.

[11] UN/CEFACT and OASIS - Enabling Electronic Business with ebXML, December 2000.

[12] Madhu Siddalingaiah, "Overview of ebXML", Sun Microsystems, August 2001.

[13] Haruo Hayami, Masashi Katsumata, Ken-ichi Okada, "Interworkflow: A Challenge for Business-to-Business Electronic Commerce", *Workflow Handbook 2001*, *WfMC*, October 2000.

[14] Jae-Gak Hwang, Se-Won Oh, ,Yong-Jun Lee, "e-Logistics Integration System based on ebXML specifications", *Proceedings of the Conference of Korea Information Science Society*, *Vol 29*, *No 2*, October 2002.

[15] Kwang-Hoon Kim, "A Process-Driven e-Business Service Integration System and Its Application to e-Logistics Services", *Proceedings of the WISE Conference*, *Lecture Note in Computer Science*, *Vol 3806*, pp. 669-678, 2005.

[16] Kwang-Hoon Kim, "A Process-Driven Interorganizational Choreography Modeling System", *Proceedings of the OTM Workshop*, *Lecture Note in Computer Science*, *Vol 3762*, pp. 485-494, 2005.

[17] The JAVATM 2 Enterprise Edition Developer's Guide, Version 1.2.1, Sun Microsystems, May 2000.

[18] Enterprise JavaBeans Programming, Revision A.1, Sun Microsystems, May 2000.

Kwanghoon Kim is an Associate Professor of Computer Science Department and Director of the Collaboration Technology Research Laboratory at Kyonggi University, South Korea. Also, he is in charge of the director of the contents convergence software research center that has been established at just summer of 2007 as a new GRRC project funded by the Gyeonggi Provincial Government, Republic of Korea. He received B.S. degree in computer science from Kyonggi University in 1984. And he received M.S. degree in computer science from Chungang University in 1986. He also received his M.S. and Ph.D. degree from the computer science department of University of Colorado at Boulder, in 1994 and 1998, respectively. He had worked as researcher and developer at Aztek Engineering, American Educational Products Inc., and IBM in USA, as well as at Electronics and Telecommunications Research Institute (ETRI) in South Korea. In present, he is a vice-chair of the BPM Korea Forum, a country-chair (Korea) and a ERC vice-chair of the Workflow Management Coalition. He has also been on the editorial board of the journal of KSII, and the committee member of the several conferences and workshops. His research interests include groupware, workflow systems, BPM, CSCW, collaboration theory, Grid/P2P distributed systems, data warehousing and mining, and process-aware information systems.

Ilkyeun Ra is an assistant professor in the Dept. of Computer Science & Engineering and Director of the Decent (Distributed Computing and Networking) Lab at University of Colorado at Denver and Health Sciences Center, Denver USA. He received the B.S. and M.S. degrees in computer science from the Sogang University in 1985 and 1987, respectively. And he also received the M.S. degree from the computer science department of the University of Colorado at Boulder in 1994, and the Ph.D. degree from the computer science department of the Seracuse University in 2001. His research interests include developing adaptive distributed system software and high speed communication system software to support High Performance Distributed Computing Applications.