From: AAAI Technical Report SS-00-03. Compilation copyright © 2000, AAAI (www.aaai.org). All rights reserved.

Towards the Cross-organizational Work Process Coordination and Enactment

Yoshihiro Masuda, Toshiya Yamada, Yuichi Ueno, Katsunori Horii

Corporate Research Center
Fuji Xerox, Co., Ltd.
430 Sakai, Nakai-machi, Ashigarakami-gun,
Kanagawa, 259-0157 JAPAN
{masuda@mb3, toshiya@mb4, yueno@mb7, khorii@mb5}.crl.fujixerox.co.jp

Abstract

The enormous growth of the Internet and the World-Wide Web led to the emergence of various forms of cooperation beyond organizational boundaries. Also, the growth of the solutions business market encouraged enterprises to develop customer-centric services, which require rapid and dynamic service process coordination for on-demand services providing. Through the observation of these tendencies, we can foresee the increase of costs to coordinate business processes beyond organizational boundaries.

This paper presents a work process representation model and computation architecture for cross-organizational work process coordination. The model realizes asynchronous work process coordination in the distributed network environment. Also, this model provides decentralized architecture for flexible work process enactment. Additionally, we implemented a prototype system called "ProcessNavigator" that is based on the above model, and we found that this model realizes the work process coordination which will allow users to register their activity without requiring explicit attention to maintaining consistencies with other activities.

1. Introduction

The advent of the Internet and the World-Wide Web has led to radical changes in business. Boundaries are being softened, and forms of cooperation across organizational boundaries have emerged. This kind of cooperation is known as "Business-to-Business (B2B) electronic commerce". Also, the increase of small-office/home-office (SOHO) workers and the diffusion of strategic outsourcing will support the growth of cross-organizational business processes over the Internet.

Another change in business that we have observed is the growth of the solutions business. Many enterprises are now changing their management policy from a technology-centric approach to customer-centric one, and are recognizing the importance of delivering one-to-one services. This change requires on-demand work process coordination for providing on-demand services that meet the customer's diverse requirements.

Realizing these changes require technology to support cross-organizational process coordination. Such coordination technology must be effective at finding new value-chains in the B2B electronic commerce, and also be effective at providing enterprise-wide flexible on-demand service. Finally, the resulting coordinated business processes can be used as "shared knowledge," to support the development of new business processes.

This report presents a framework for cross-organizational work process management. The model and the architecture we introduce provide asynchronous work process coordination and decentralized work process management in a distributed network environment.

The remainder of this report is organized as follows: Section 2 presents related work on distributed workflow management and automatic workflow coordination. Section 3 summarizes features and issues for realizing cross-organizational work process coordination. Section 4 presents the work process representation model and the mechanism for realizing asynchronous work process coordination. Section 5 presents decentralized work process enactment architecture. Section 6 describes the design and implementation of our prototype system, *ProcessNavigator*. Section 7 offers concluding remarks and future work.

2. Related Work

It had been a common understanding that the benefit of introducing a work process management system would reduce the cost of routing jobs. Research on workflow management systems that was started in the late 1980's focused on the flexibility of workflow definition and workflow enactment control. However, triggered by the growth of the Internet and the World-Wide Web, current general interests in the workflow research field are shifting toward cross-organizational workflow for amplifying the value-chains in the Internet. The rest of this section summarizes the research on distributed workflow and automatic workflow coordination.

2.1 Distributed Workflow Management

Some of the recent papers deal with the aspect of

Copyright © 1999, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

distribution in workflow management. We can identify three approaches in them. The first approach is *shared distributed work process definition.* This approach allows all the workflow participants to define a shared work process in distributed manner. Examples of this approach include Task Manager[14], ActionWorkflow[18], Regatta[21], ContAct[7], and BSCW-Flow[15]. The second approach is *inter-workflow management system federation.* In this approach, workflow engines are distributed in the network, and controls for enactment are exchanged among the engines. GroupFlow[19] allows several enactment engines to coordinate distributed workflow data on a shared directory server. Also, in 1994 the Workflow Management Coalition was founded to define standards for terminology and interfaces of workflow management systems[21]. This standard defines the interface for federating multiple workflow engines, and several products were released from software vendors. The third approach is *work-packet circulation.* This approach circulates documents or folders with attached workflow data. A number of systems built on e-mail have been proposed, and there are also some current attempts to embed workflow data in HTML and XML documents[10].

2.2 Automatic Workflow Coordination

The purpose of automatic work process coordination in previous work was focused on extracting optimal consistent workflow data in terms of resolving the local constraints between activities. Additionally, existing methods are based on the IPO (Input-Process-Output) workflow representation, and the problem addressed was resolving the constraints on execution order.

We can identify three approaches in automatic work process coordination. The first approach is *aggregated coordination.* In this approach, constraints specified as rules are gathered in a coordination server, and the coordination server extracts consistent workflow data. Examples of this approach include WebFlow[8], Panta Rhei[6], and $TriGS_{flow}$[13]. The second approach is *negotiation in a shared computing space.* Autonomous agents are used in this approach, and workflow data are extracted through communication and negotiation in a shared data space. A typical example of this approach is ADEPT[11]. The third approach is *mobile coordination agent.* Mobile agents search activities over the network, and organize workflow data. DartFlow[3] and Mobile Agent-X[4] are examples based on this approach.

3. Natures of Cross-organizatinal Work Process Coordination

Existing intra-organizational workflow management systems handle work processes that can be maintained in a single (usually proprietary) server on a local area network. By contrast, managing cross-organizational work processes on the Internet bring up issues that are peculiar to open systems.

3.1 The Internet as an Open System

The Internet is thought of as an open system in that it has no central management. From the attributes of open systems listed by C.Hewitt [9], we recognized that *concurrent, asynchronous, decentralized* management is required for realizing a cross-organizational workflow management system.

Work Process Representation for Open Work Process Coordination. Previous work on automatic workflow coordination is based on the notion that work processes can be coordinated through constraint resolution among activities. Usually, the constraints are specified as rules on inputs/outputs or pre/post-conditions between activities.

However, if users are allowed to add, delete, or modify the rules in a concurrent, asynchronous, decentralized manner, it is difficult to identify the subject rule set for computing. Accordingly, we conclude that rule-based reasoning or constraint resolution show some difficulties for cross-organizational automatic work process coordination.

In addition, the above existing method brings other drawbacks. Users have to ensure that consistency is kept between existing rules when they intend to define rules for an activity. In a situation where rules are added, deleted, or modified in a concurrent, asynchronous manner, this situation will worsen.

The above implies that it requires coordination from the bottom-up, with local constraint resolution and a common specification for representing dependencies between activities.

Decentralized System Architecture. Existing server-centric architecture brings heavy centralization of computing load. In general, a workflow management system consists of three primary functional components: *work process definition*, *work process enactment*, and *service providing*. These components are likely to be used by different kinds of users widely distributed in the network. In this sense, a highly distributed architecture will be realized by decomposing the systems into these components in the distributed network environment.

3.2 Commitment as Role Assignment

Existing workflow management systems have been designed based on the notion that workflow participants are assigned to activities based on their authorities or roles. However, in the case of cross-organizational work process management, we identify difficulties in assuming someone has the authority for assigning participants to activities. This implies that the role assignment in ordinary intra-organizational workflow management should be replaced by making commitments among workflow participants in the case of cross-organizational workflow management systems.

In addition, while service providers and consumers are treated as distinct entities within ordinary workflow management systems, they should be treated as heterogeneous in cross-organizational work process management, because in B-to-B electronic commerce the

situation may exist in which service providers casually delegate parts of their activities to other service providers.

4. Distributed Work Process Coordination

As we discussed above, nearly-existing automatic work process coordination is accomplished by resolving constraints on their execution order. The relations extracted in this way constitute a network structure, and this makes it difficult to modify the overall structure because the relations are tightly connected to each other (N-to-N connection), and a partial change affects consistencies as the whole.

Through the above observation, we use task/subtask relationships to represent dependencies between activities. This represents decomposed goals rather than execution orders, and it constitutes a tree structure (1-to-N connection). This nature provides flexibility of modifications because the relationship between activities can be freely inserted or deleted without any changes on sub-structures. Although this nature is effective for realizing asynchronous work process coordination in the distributed network, this raises hard new issues about representing the semantics of activities.

4.1 Representing Semantics of Activities

Using task/subtask relationships to represent dependencies between activities requires interface description for representing semantics of activities, because work process coordination is done through matching consumers' requirements and providers' services.

We found a cue to solving the above issue from ProcessHandbook[17]. ProcessHandbook is a repository of activities, and huge numbers of activities are extracted from almost one thousand work processes in the field and are classified in an abstraction hierarchy. The hierarchy is structured with eight specialized fundamental verbs: *create, destroy, modify, preserve, combine, decompose, decide,* and *manage*. Based on the this analysis, we were led to the following hypothesis,

> *The operations to represent semantics of activities are finite, and the diversity of the semantics is brought by combination of subjects to be operated.*

Then we introduced an activity description that consists of an operation name and types of subject. (This is formalized as in Figure 2 (1).)

Here, the operation name is specified as a verb from the eight fundamental operations that ProcessHandbook provides. Each operation has own argument patterns, and it applies common sense to the semantics of activities as illustrated in Figure 1.

Using the above representation, we can describe diverse kinds of activities in the fields. Table 1 shows examples specific to the above representation.

4.2 Representing Dependencies between Activities

To represent task/subtask dependencies, each activity consists of three elements: *activity name*, *activity interface,* and *subtask*. Activity name is an identifier of an activity the service provider provides. Activity interface is specified as an activity description we introduced in above, and it is used for matching with subtask description in other activities. Subtask consists of activity descriptions, and they represent the external activities delegated from the activity.

When the activity is invoked from a work process executed by users, a "token" which represents an incoming request to the activity is stored in the queue, then one or more instances of the subtask are created, and they are associated with the token. The work process instances are owned and executed by the service provider. (This is formalized as in Figure 2 (2).)

4.3 Work Process Coordination

Two types of computation are applied for decentralized work process coordination. Both of these computations are applied in locally.

The first type is the computation for resolving task/subtask relationships, and it is done through matching the subtask specification and the activity interface specified in the registered activities. Here, equality of the operation name and the types of subjects are examined and associated with each other. As we will present later, activity interfaces for the activities provided are registered and indexed in the activity directory, and each service server searches activity interfaces which correspond to the descriptions in the subtask.

The second type is the computation for extracting composite activities from types of subject(s). In open systems, almost infinite numbers of possible combinations of activities can be extracted. This implies that new composite services such that humans cannot find will be found through matching the types of input subject and output subject. This type of coordination is done by recursively applying the built-in composition rules illustrated in Table2. In addition, the extracted composite services are registered into the activity directory, and made available for the task/subtask coordination. (This is formalized as in Figure 2 (3).)

5. Decentralized Work Process Management Architecture

The work process representation model we introduced in the previous section enables highly distributed architecture for work process management.

5.1 Architecture Overview

The system architecture is composed of three functional elements distributed in the network: *activity directory*, *activity server*, and *enactment engine*. The activity directory is implemented as a server which maintains a shared index of activity interfaces. When an activity interface is registered into the activity directory, it is indexed and made accessible from other activities.

The activity servers are servers owned by a service provider, and they provide services that their activity interface indicates. As we mentioned in the previous section, each activity server has the token queue for preserving incoming tokens from the enactment engines.
Users control the execution of work process instances with the enactment engine. Each activity server has an enactment engine, and it is used for managing the execution of the work processes instantiated from the subtask description. Figure 3 illustrates the overview of the architecture.

5.2 Decentralized Work Process Enactment

The architecture we introduced above is highly distributed, and the execution of work process is done through exchanging tokens among activities. Each token consists of identifier, parent token, child tokens, and status. Tokens maintained in a work process instance form a tree structure along with the work process execution, and their relations are held as the parent token and child tokens. The status of the activity the token is held as the status. (This is formalized as in Figure 2 (4).)
When an activity is invoked from the process instance, the user is in execution. A new token derived from the current token is created and sent to the corresponding activity. When the work process instance in the activity terminates, the activity sends back the token to the parent token with the parent address. The parent token receives the child tokens and examines the current status, then determines their actions. As a result of the control we presented here, an execution chain of the activities is formed per each work process instance.

6. Prototype System

We have finished the first implementation of the prototype system called "*ProcessNavigator*," which provides asynchronous work process coordination in a distributed network environment.
ProcessNavigator is implemented on a Java platform (JDK1.1.7), and their server-client architecture is realized on the Java RMI framework. The activity directory is the activity repository and it provides work process coordination service. Client applets are downloaded through the network and run on standard Web browsers.

Activity Directory. The activities registered by users are indexed and stored in the activity directory. The activity directory provides information about registered activities to the service servers and end-users when they break down a task into subtasks.

User Interface. Two kinds of applets are provided in ProcessNavigator. The service provider's applet enables users to register the service they provide. The other one is the applet designed for end-users, and it provides dynamic work process decomposition and service selection. Figure 4 illustrates a screen shot of the user interface for the service provider.

Through the trial use of the interface, we found that it enables users to register their activity without paying attention, to keep global consistencies between other activities.

7. Conclusion

This report presented a work process representation for cross-organizational work process coordination in a distributed network environment, and we also presented the system architecture for decentralized work process management. We introduced a new method for representing semantics of activity. Each activity is described with object-oriented notation, and dependencies between activities are represented as a task/subtask relationship. With this method, modification of subtask specification in an activity can be accomplished without any changes of the subtask specification in other activities. This allows users to register activities asynchronously in the distributed network.
The above model realizes highly distributed architecture for work process management. The functional elements of the architecture: *activity directory*, *activity server*, and *enactment engine* are distributed in networks, and flow management is accomplished by exchanging tokens.

Application Domain. Currently we are assuming two kinds of application domains. One is an infrastructure for web-based on-demand service. Future on-demand service will require dynamic process coordination for providing services that meet the client's diverse requirements, and in such situations, the above technology will provide a unique mechanism for dynamic service extraction and incremental service organizing. Also, sharing and utilizing of the work processes thought of as "organizational knowledge" will be helpful for developing new services. Another application domain we can foresee is B-to-B coordination service for SOHO workers. Presenting candidate work processes organized between other workers will amplify business collaboration on the Internet.

Future Work. Further improvements on work process representation will be required for applying our model into practical systems, and we are now investigating some extensions to our model. These include the introduction of control structures, meta-activity, dynamic participant assignment, dynamic shared resource assignment, synchronization among work processes, and so on.

Also, we are now investigating the emergence of work processes in the Internet. The work process representation we presented in here is highly distributed and suitable for computing semantics of activities in an open network environment, and we believe that the emergence of work processes can be realized by applying combinatorial optimization methods such as Genetic Programming into our model.

Acknowledgement

The authors wish to thank our colleagues at FX CareerNet, subsidiary company of Fuji Xerox, for their thoughtful commentary. We could get many insights about applicability of our system in real use though the discussions with them. Also we wish to thank J. Maxson and L. Klotz for their helpful comments on this paper.

References

[1] G. Alonso, C. Mohan, R. Gunthor, D. Agrawal, A. El Abbadi, M. Kamath: "Exatica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management", In Proceedings of IFIP Working Conference on Information Systems for Decentralized Organizations, 1995

[2] B. Berkem: 'Contextual Objects' or 'Goal-Oriented Business Components' for the Business Process Modeling, ACM OOPSLA'98 Workshop on Implementation and Application of Object-Oriented Workflow Management Systems, 1998

[3] T. Cai, P.A. Gloor, S. Nog: "DartFlow: A Workflow Management System on the Web using Transportable Agents", Technical Report of Dartmouth College (PCS-TR96-283), 1996

[4] T-H. Chia, S. Kannapan: "Strategically Mobile Agents", DRI Working Paper, 1998 http://willow.tc.cornell.edu/pub/People/kannapan.html

[5] W.H. Davidow, M.S. Malone: The Virtual Corporation, HarperCollins, 1992

[6] J. Eder, H. Grois, W. Libhart: "The Workflow Management System Panta Rhei", In A.Dogac (Eds.) Workflow Management Systems and Interoperability, Springer-Verlag, 1998 http://www.ifi.uni-klu.ac.at/

[7] J. Galler, J. Hagemeyer, A.-W. Scheer: "The Coordination of Interdisciplinary Teams in Workflow Projects", In Proceedings of the Third Interdisciplinary Information Management Talks (IDMT'95), 1995 http://www.iwi.uni-sb.de/contact/cont_21.html

[8] A. Grasso, J. Meunier, D. Pagani, R. Pareschi: "Distributed Coordination and Workflow on the World Wide Web", In Computer Supported Cooperative Work: The Journal of Collaborative Computing 6, pp.175-200, 1997

[9] C. Hewitt "Offices are Open Systems", ACM Transactions on Office Information Systems, Vol.4 (3), pp. 271-283, 1986

[10] IEEE Internet, "Workflow Protocols for the Web", IEEE Internet, July/August, 1998

[11] N.R. Jennings, P. Faratin, M.J. Johnson, P. O'Brien, and M.E. Wiegand, "Using Intelligent Agents to Manage Business Processes", In Proceedings of the First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96), pp. 345-360, 1996

[12] M. Kamath, K. Ramamritham, N. Gehani, D. Lieuwen: "WorldFlow: A System for Building Global Transactional Workflows", In Proceeding of Seventh International Conference on High Performance Transaction Systems (HPTS'97), 1997 http://www-ccs.cs.umass.edu/kamath/hpts97.html

[13] G.Kappel, S.Rausch-Schott, W. Retschitzegger: "Coordination in Workflow Management Systems - A", In W. Conen, G. Neumann eds.) Coordination Technology for Collaborative Applications – Organizations, Processes and Agents", LNCS-1364, Springer, 1998

[14] T. Kreifelts, E. Hinrichs, G. Woetzel: "Sharing To-Do Lists with a Distributed Task Manager", In Proceedings of ECSCW'93, Kluwer Academic, pp.31-46,1993

[15] T. Kreifeltsm E. Hinrichs, G. Woetzel: "BSCW-Flow: Workflow in Web-based Shared Workspaces", Proceedings of the Workshop on Cross-Organizational Workflow Management and Co-ordination http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-17/

[16] R.J. Laubacher, T.W. Malone, the MIT Scenario Working Group: "Two Scenario for 21st Century Organizations: Shifting Networks of Small Firms or All-Enconpassing "Virtual Countries" ?", 1997 http://ccs.mit.edu/21c/21CWP001.html

[17] T.W. Malone, K. Crowson, J. Lee, B. Pentland, C. Dellarocas, G. Wyner, J. Quimby, C. Osborne, A. Bernstein: "Tools for Inventing Organisations: Toward a Handbook of Organizational Processes", 1997 http://ccs.mit.edu/CCSWP198/

[18] R. Medina-Mora, T. Winograd, R. Flores, F. Flores: "The Action Workflow Approach to Workflow Management Technology", In Proceedings of ACM CSCW'92, pp.281-288, 1992

[19] L. Nastansky, W. Hilpert: "The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automation", In Wolfinger, Bernd (Ed.): Proceedings of 24th Anual Conference of the German Computer Society during 13th World Computer Congress, IFIP '94, Springer Verlag, Berlin, Heidelberg etc., pp. 473 -479, 1994

[20] G. Riempp: Wide Area Workflow Management – Creating Partnerships for the 21st Century, Springer Verlag, 1998

[21] K.D. Swenson: "Visual Support for Reengineering Work Process", In Proceedings of ACM COOCS'93, pp.130-141, 1993

[22] WfMC: "WfMC: Reference Model – The Workflow Reference Model" (WFMC-TC-1003, 1.1), 1994

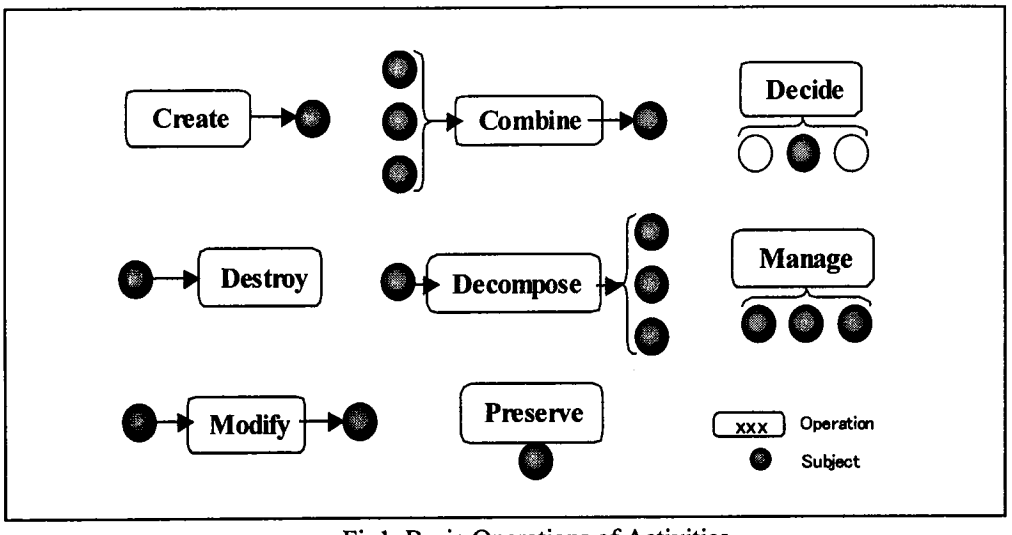

Fig1. Basic Operations of Activities

Table1. Examples of Activity Representation

Category	Activity	Operation and Subjects
Create	Design a web page Plan an event Write a book	**Create**→Web-Page **Create**→Event **Create**→Book
Destroy	Dispose a document	Document→**Destroy**
Modify	Transport a parcel Translate a document Convert a file Proofread a document Scan a document Print a document	Parcel(*source*)→**Modify**→Parcel(*destination*) Document(*"English"*)→**Modify**→Document(*"Japanese"*) Document(*"Word"*)→**Modify**→Document(*"PDF"*) Document(*"Draft"*)→**Modify**→Document(*"Proofread"*) Document(*"Paper"*)→**Modify**→Document(*"Digital"*) Document(*"Digital"*)→**Modify**→Document(*"Paper"*)
Combine	Assemble a PC Edit a magazine Calculate a bill Make an archive file Indexing documents	{CPU, IO-Board, HDD}→**Combine**→PC {Article, Advertise}→**Combine**→Magazine {Data}→**Combine**→Bill {File}→**Combine**→Archive {Document}→**Combine**→Indexed-Document
Decompose	Decompose an archive file	Archive→**Decompose**→{File}
Preserve	File documents Reserve a seat	**Preserve**{Document} **Preserve**{Seat}
Decide	Give assurance Search in a database Approve a document	Document→**Decide**→Document(*"Assured"*) Database→**Decide**→Data Document→**Decide**→Document(*"Approved"*)
Manage	Manage budget Manage personnel Manage schedule	**Manage**[Budget] **Manage**[Person] **Manage**[Schedule]

Table 2. Activity Composition Rules

Pattern	Interpretation
Create(out:X) + Modify(in:X, out:Y)	Create(out:Y)
Modify(in:X, out:Y) + Modify(in:Y, out:Z)	Modify(in:X, out:Z)
Combine(in:*, out:X) + Modify(in:X, out:Y)	Combine(in:*, out:Y)
Modify(in:X, out:Y) + Decompose(in:Y, out:*)	Decompose(in:X, out:*)

(1)

$ActivityDescription = \{(operation, arguments) \mid operation \in OperationSet, arguments \subseteq Argument\}$

$OperationSet = \{$create, destroy, modify, combine, decompose, preserve, decide, manage$\}$

$Argument = \{(subjectTypes) \mid subjectTypes \subseteq SubjectType\}$

$SubjectType = \{$type name of subjects$\}$

(2)

$Activity = \{(activityName, activityInterface, subtask, processInstances, incomingRequests) \mid activityName \in String, activityInterface \in ActivityDescription, subtask \subseteq ActivityDescription, processInstances \in ProcessInstance, incommingRequests \in Token\}$

(3)

$CompositeActivity = \{(activityInterface, activities) \mid activityInterface \in ActivityDescription, activities \subseteq Activity\}$

(4)

$Token = \{(id, parent, children, status) \mid id \in TokenIdentifier, parent \in ParentToken, children \subseteq ChildToken, status \in Status\}$

$TokenIdentifier = \{$unique identifiers in the system$\}$

$Status = \{$waiting, in_process, complete, aborted, suspended$\}$

$ParentToken = \{(pid, address) \mid pid \in TokenIdentifier, address \in TokenAddress\}$

$ChildToken = \{(cid, status) \mid cid \in TokenIdentifier, status \in Status\}$

$TokenAddress = \{$unique address in the system$\}$

Fig2. Formal Definition of the Model

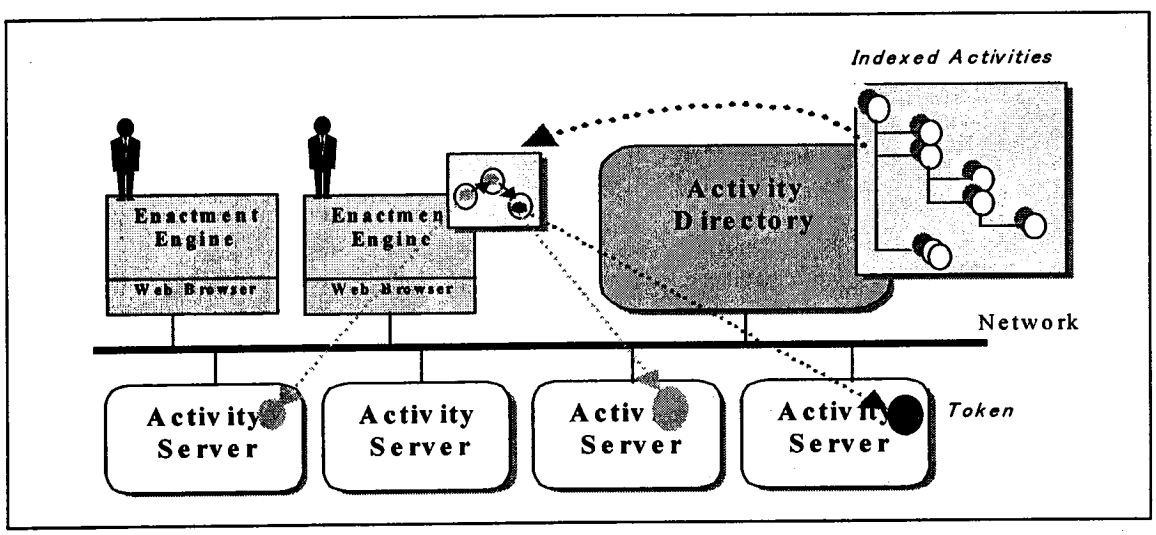

Fig3. Decentralized Work Process Enactment Architecture

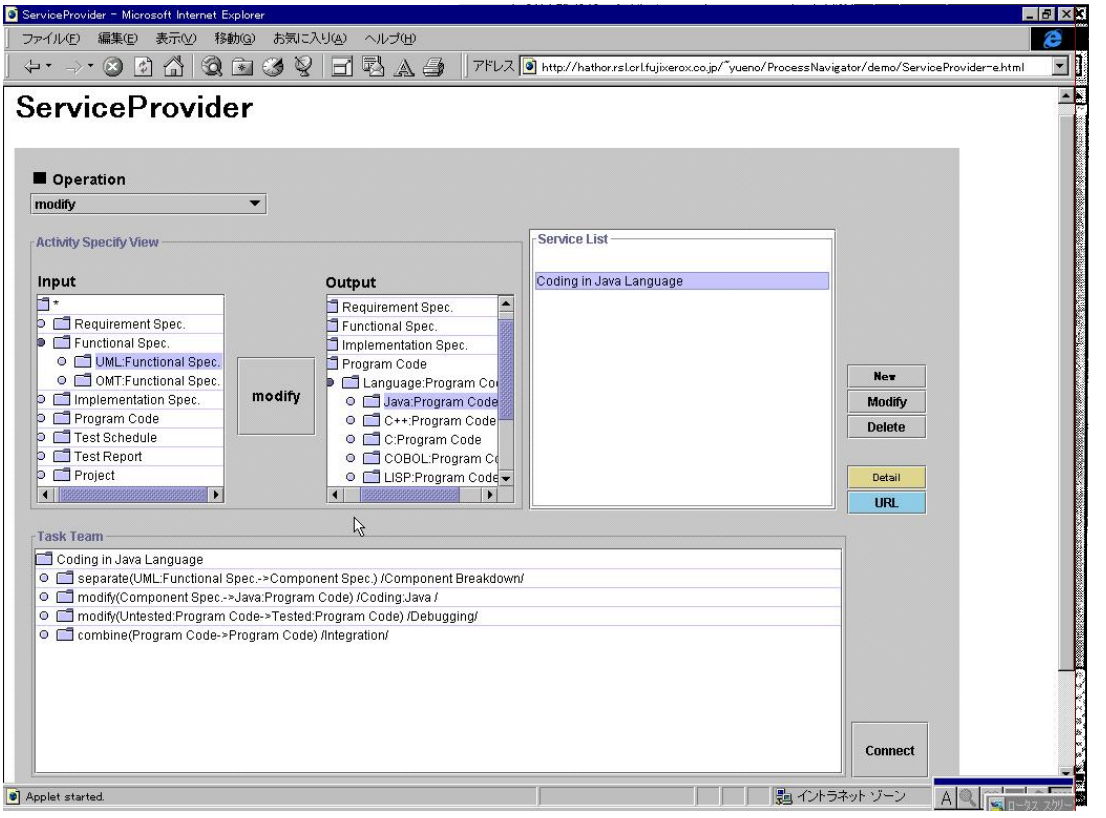

Fig4. Screen Shot of ProcessNavigator