Achieving Dynamic Inter-organizational Workflow Management by Integrating Business Processes, Events, and Rules*

Jie Meng, Stanley Y. W. Su, Herman Lam and Abdelsalam Helal

Database Systems R&D Center

Department of Computer and Information Science and Engineering

University of Florida, Gainesville, FL 32611-6125, USA

Telephone: 352-392-2680

{jmeng,su,hlam,helal}@cise.ufl.edu

Achieving Dynamic Inter-organizational Workflow Management by Integrating Business Processes, Events, and Rules*

Abstract

In the highly competitive and rapidly changing and expanding global marketplace, business organizations often form a virtual enterprise with others to achieve various business goals. Traditional workflow technology is used to coordinate the activities of an organization. Since the business environment of a virtual enterprise is highly dynamic, it is necessary to develop a workflow management technology that is capable of handling dynamic workflows across enterprise boundaries. This paper describes the development of a dynamic workflow model and a dynamic workflow management system for modeling and controlling the execution of inter-organizational business processes in a virtual enterprise environment. The dynamic inter-organizational workflows are used to integrate e-services provided by the participating organizations. The dynamic workflow model (DWM) described in this paper enables the specification of dynamic properties associated with a business process model. It is an extension of the underlying model of the Workflow Management Coalition's Workflow Process Definition Language by adding connectors, events, triggers and rules as modeling constructs, encapsulating activity definitions, and supporting e-service requests. We also introduce a constraint-based, dynamic service binding mechanism for the enactment of business processes. The technique for run-time modifications of the process models to alter the course of executing workflow instances is also addressed in our work. The dynamic workflow management system is a part of an information infrastructure being developed for supporting Internet-based Scalable E-business Enterprise (ISEE).

** This project is supported by the National Science Foundation (grant number EIA-0075284).*

1 Introduction

The rapid growth of the Internet, together with the increasingly competitive and expanding global marketplace, gave birth to e-business. To achieve a common business goal, different organizations often need to form virtual alliances to address rapidly changing market conditions in a more flexible and effective way than ever before.

Workflow technology is an enabling technology for coordinating the activities of an enterprise. It allows people and companies to model business processes and to control the execution of these processes. Most commercial workflow management systems are used to manage the workflows of individual organizations. They are centrally controlled and use a static workflow model to specify business processes. Some of them support dynamic changes to the defined process models. However, the dynamic properties they provide are very limited.

E-business is a new application area that can benefit from the workflow technology. The workflow of a virtual enterprise is an inter-organizational workflow since it requests services provided by more than one organization. Unlike a traditional workflow management system, which handles business processes within a single organization, an inter-organizational workflow management system allows inter-organizational processes to be executed over heterogeneous systems distributed across the Internet. Because of the dynamic nature of e-business, the workflow management system that supports inter-organizational business processes must be dynamic; i.e., able to accommodate the changing business policies and strategies of participating organizations, handle expected or unexpected events, and support run-time modifications to process models. Business processes also need to be integrated with business events and business rules so that business constraints, policies, strategies, and regulations specified by rules can be enforced upon the occurrences of some events during the execution of a business process. Also, the participating organizations of a virtual enterprise have different resources (i.e., data, application systems, people, etc.) and provide manual and automated services for the manipulation and access of these resources. Since business organizations can enter and leave the Internet world freely, their memberships in a virtual enterprise and their services may change from time to time. The dynamic nature of services and service providers requires that service requests specified in a process model be dynamically bound to services and their providers at the time when an instance of the process model is in execution. To support e-business over the Internet, another important requirement is that software systems developed for managing business processes, events and rules must be scalable.

In this paper, we describe a dynamic inter-organizational workflow solution to meet the above requirements. Our idea is to use dynamic inter-organizational workflows to integrate distributed and heterogeneous resources and services of the business organizations that participate in a virtual enterprise. We propose a dynamic workflow model (DWM) to enable the specification of dynamic properties associated with a business process model. The DWM is an extension of the underlying model of the WPDL [WfMC99] by adding event, trigger and rule to WPDL's modeling constructs. By doing so, we integrate business events and business rules with business processes. The enactment of a business process can post events to trigger the processing of business rules and the processing of business rules may in turn enact business processes. Also, the processing of rules may dynamically alter the process model at run-time. We also separate control information (i.e., Split, Join) from activity definition so that each activity definition is encapsulated and reusable. In this work, we treat all the sharable tasks performed by people or automated systems in a virtual enterprise as electronic services or *e-services*. E-service requests are specified in each activity definition of a process model according to some *standardized e-service templates* and are bound to the proper service providers at run-time by using the services of a broker to identify the suitable providers. By doing this, process models are separated from (i.e., not bound to) the specific service providers when they are defined. Changes in the membership of a virtual enterprise (i.e., its service providers) will not affect the process models, affect only the binding. The integration of business processes with business events and rules and the dynamic binding of e-services to service providers give the workflow management system its dynamic properties.

The organization of this paper is as follows. In Section 2, research related to workflow and virtual enterprise is surveyed. In Section 3, the global architecture of the dynamic workflow management system is introduced. The e-service specification, dynamic workflow model (DWM), and our extensions to WPDL are explained in Section 4. Section 5 gives a sample scenario. The design and implementation of the key components of the system are given in Section 6. Section 7 summarizes our research and contributions.

2. Related Work

2.1 WfMC and WPDL

The Workflow Management Coalition (WfMC) was founded in August 1993. Its mission is to promote and develop the use of workflow through the establishment of standards for software terminology, interoperability, and connectivity between workflow products. The Workflow Reference Model has been developed from the generic workflow application structure by identifying the interfaces within this structure, enabling products to inter-operate at a variety of levels [WfMC95]. The Coalition has developed a framework for the establishment of workflow standards based on this reference model. In the framework, Interface 1 includes a common meta-model for describing the process definition and a textual grammar for the interchange of process definitions (Workflow Process Definition Language – WPDL) [WfMC99]. It focuses on the specification of a process definition meta-model. This meta-data model identifies the basic set of entities and attributes used in the exchange of process definitions. In this model, a workflow process contains one or more activities. The activities are associated with workflow applications and workflow participants. Transitions that connect these activities together determine the control flow in this workflow process. Transitional conditions can be defined to identify the flow or execution conditions. A variety of attributes describe the characteristics of this limited set of entities. Based on this model, vendor specific tools can transfer models via a common exchange format. In our work, we extend the underlying meta-model of WPDL by adding events, rules and connectors to its set of modeling constructs.

2.2 Related Workflow Research

There are many research efforts in the workflow area. [Alon95] [Ceri97] [Sheth97] [Stri98] [Kim00] [Lazc00]. In this section, we review the research related to our work.

Events and rules have been used in several research projects. An example is the WIDE project [Ceri97]. WIDE uses a distributed architecture for workflow management, based on a database management system with active rule supports. WIDE's model specification allows the definition of Event-Condition-Action (ECA) rules to support exceptions and asynchronous behavior during workflow enactment. WIDE also provides transactional support to workflow management. Events and rules are also used in the EvE project [Gepp98] as the fundamental concepts for defining and enforcing workflow logic.

A distributed ECA rule-based enactment architecture is also investigated in EvE. Unlike WIDE and EvE, the dynamic workflow model described in this paper integrates business events and rules to capture dynamic properties of business processes. Synchronous events are access points of a process model where organizations can attach business rules to adapt to a changing business environment. Asynchronous events can be posted within a process model to notify interested organization of some processing milestones of an enacted business process or exceptions.

A dynamic workflow management system needs to be able to modify a process model at run-time to adapt to dynamic business conditions and exception situations. There are some research efforts in this area. Most of them deal with the dynamic changes of process models used in traditional workflow systems. [Reic98] presents a formal foundation for supporting dynamic structural changes of running workflow instances. Based upon a formal WF model (ADEPT), a complete and minimal set of change operations (ADEPT_flex) is defined to support users in modifying the structure of a running WF, while maintaining its (structural) correctness and consistency. [Mull99] describes a rule-based approach for the detection of semantic exceptions and for dynamic workflow modifications, with a focus on medical workflow scenarios. Rules are used to detect semantic exceptions and to decide which activities have to be dropped or added. Different from the existing works, we address run-time modifications to inter-organizational process models in our work.

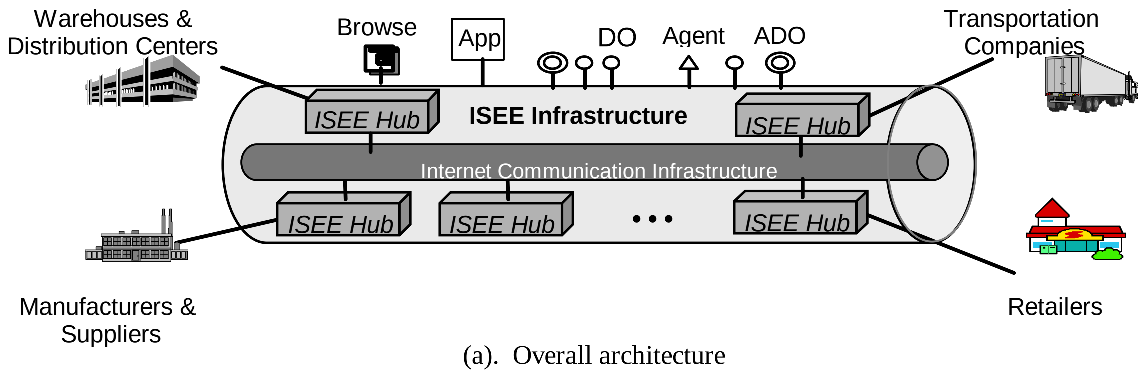
Recently, the use of the workflow technology to manage e-businesses and virtual enterprises has drawn a lot of attention in the academic community. Several research projects attempt to tackle workflow management problems in these new application areas. Here, we describe two representative projects. WISE (Workflow based Internet SErvices) is a project conducted at the Swiss Federal Institute of Technology [Lazc00] [Alon99]. It aims to design, build, and test a commercially viable infrastructure for developing distributed applications over the Internet. The infrastructure provides an Internet-based workflow engine acting as the underlying distributed operating system for controlling the execution of distributed applications, and a process modeling tool for defining and monitoring the process. CrossFlow is a European research project for supporting cross-organizational workflow management in virtual enterprises [Cross00]. Its goal is to develop and implement a mechanism for connecting WfMS and other WfMS-like systems of different organizations in cross-organizational workflows and electronic commerce settings. CrossFlow defines a service-oriented model for cross-organizational workflows. In their service-oriented model, a service specifies which part of workflow it fulfills. For an external service, service selection at run-time will be based on the QoS parameters given in service specifications. A flexible change control mechanism is also introduced in CrossFlow to react to potential problems during a workflow execution [Gref98].

Our inter-organizational workflow system integrates e-services provided by participating organizations of a virtual enterprise. Unlike the service definition in CrossFlow, the e-services in our inter-organizational workflow system are defined and provided independent of business process models. E-service requests are bound to particular service providers at run-time through dynamic service binding. The dynamic workflow management system we describe in this paper supports more dynamic properties than that of other inter-organizational workflow systems, including WISE and CrossFlow.

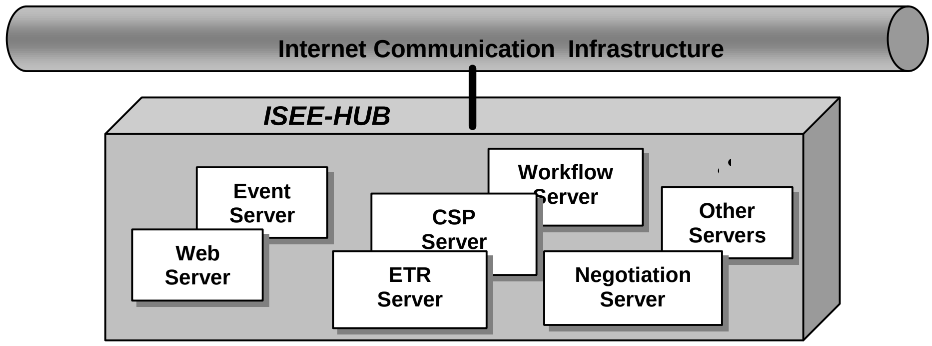
3. Architecture of the Dynamic Workflow Management System

3.1 Overview of ISEE Infrastructure

A research project supported by the National Science Foundation is being carried out at the Database Systems Research and Development Center of the University of Florida. This project is entitled "Research on Advanced Technologies to Support Internet-based Scalable E-business Enterprises (ISEE)". It aims to build an advanced information infrastructure to support collaborative e-business and other distributed applications. The ISEE infrastructure is formed by a network of ISEE hubs as shown in Figure 1(a), each of which has a number of replicable ISEE servers (Figure 1(b)) that provide ISEE-services to support the requirements of collaborative e-business. For example, the Event Service supports flexible and dynamic communication among loosely coupled systems that are distributed across organizations. The Event-Trigger-Rule Service provides timely and automated response to events. Other available ISEE-services to support collaborative e-business include the Automated Negotiation Service, Constraint Satisfaction Processor Service, etc. An important service provided by the ISEE infrastructure is the Dynamic Workflow Service. A dynamic workflow management system that implements this service will be described in the next section.



(a). Overall architecture



(b). An ISEE Hub

Figure 1. ISEE Infrastructure

3.2 Architecture of the Dynamic Workflow Management System

The global architecture of the dynamic workflow management system is shown in Figure 2. The system consists of a network of individual dynamic workflow management systems (WFMS) and a centralized Broker Server. Each WFMS is a part of an ISEE Hub, consisting of a Workflow Server, an Event Server, and an Event-Trigger-Rule (ETR) Server. The centralized Broker Server is to manage the e-service specifications that have been registered by participating business organizations, and to match e-service requests against these specifications for selecting the suitable e-service providers. A Broker Proxy is installed in each ISEE Hub to communicate with the centralized Broker Server.

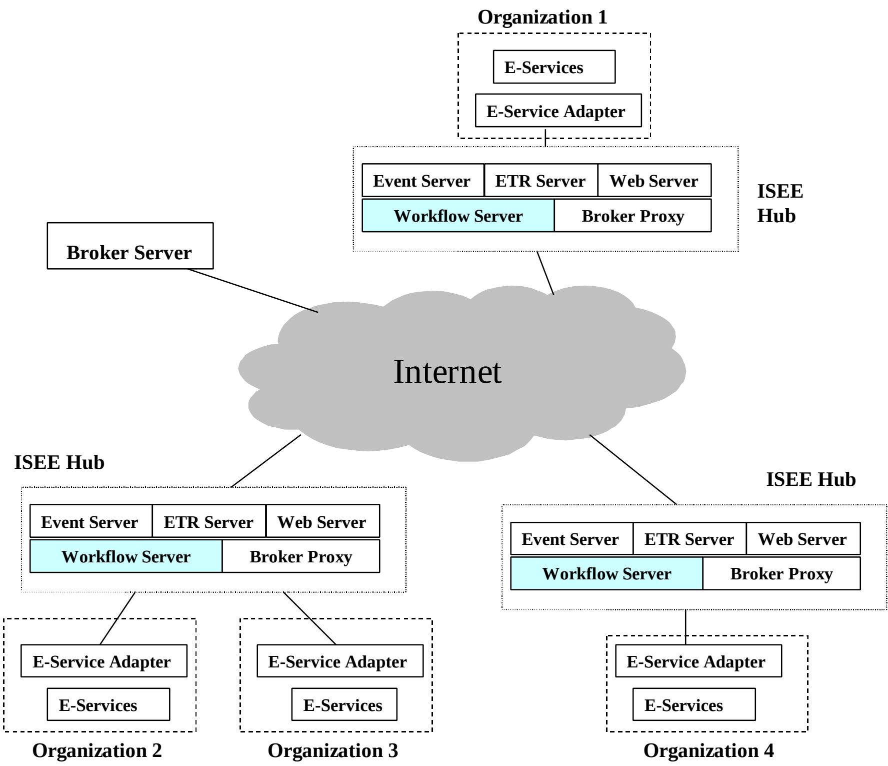


Figure 2 Global Architecture of the Dynamic Workflow System

The Workflow Server is the key component of the WFMS. It is composed of two sub-components: the Process Definition Tool and the Workflow Engine. The Process Definition Tool is for designing inter-organizational process models, which are modeled by a dynamic workflow model (DWM). An enactment of a process model is called a *workflow instance* of that model. The Workflow Engine schedules the execution of the workflow instance according to the process model specification. During the execution, the Workflow Engine makes use of the services provided by other servers such as the Event Server, the ETR Server and the Broker Server to achieve the dynamic properties (i.e., the active, flexible and adaptive properties) of the workflow management system. The Event Server handles the incoming and outgoing events to and from an ISEE Hub. It receives the asynchronous events from the local Workflow Engine and distributes them to the remote Event Servers of the other ISEE Hubs. The ETR Server maintains/executes all of the rules managed by it. It receives events from the Event Server (for asynchronous workflow events) or directly from the Workflow Engine (for synchronous workflow events), and executes those rules that are related to the events. The relationships between events and rules are specified by triggers.

In an ISEE virtual enterprise, the inter-organizational process models are designed to capture the business processes of the entire virtual enterprise. The participating organizations of a virtual enterprise may have their own process models that are processed by their own workflow systems. These "local" process models can be enacted by the dynamic workflow management system as a part of an inter-organizational process model. The inter-organizational process models, once designed, are made accessible to participating organizations. They can be stored in a central repository and be checked out and customized by participating organizations to meet their local needs. However, for scalability reasons, we propose to replicate these inter-organizational process models as well as the Workflow Server at all the ISEE-hub sites. Authorized users of the virtual enterprise can enact a process model at any site. The workflow instance created by the enactment will then be managed by an instance of the Workflow Server at that site. Thus, concurrent workflow instances initiated at different sites are controlled and managed by multiple instances of the Workflow Servers.

Participating business organizations can perform and contribute different manual or automated tasks, which are useful for the operation of a joint business. These tasks are uniformly encapsulated as e-services and are invoked in the execution of an inter-organizational business process. An e-service adapter needs to be installed at each organization's site as a wrapper of its e-services so that they can be invoked in the execution of an inter-organizational business process.

4. Dynamic Workflow Model

In this section, we introduce a dynamic workflow model (DWM) for modeling business processes. Since process models defined in DWM may invoke manual and automated tasks that can be carried out by different organizations of a virtual enterprise, we shall first present a uniform way of specifying these two general types of tasks uniformly as e-services. We then present DWM as an extension of the underlying model of WfMC's WPDL. After that, we shall delineate the dynamic properties that a workflow management system can have if it is built upon DWM.

4.1 E-Services

In a virtual enterprise, participating business organizations can perform and contribute different manual or automated tasks, which can be specified uniformly as e-services that can be invoked by an automated system. A participating organization can provide multiple e-services and an e-service can be provided by multiple organizations. In the Internet environment, providers of e-services may change frequently; new providers are added and old providers become unavailable. In modeling business processes, it is therefore important to separate e-service requests specified in a process model from their providers. That is, a process model should not statically bind its e-service requests to specific providers at the time a process model is defined. The binding should occur at run-time when the available providers are known to the workflow management system.

In order to introduce a standard way for defining each e-service, it is useful to categorize and standardize e-services and their specifications by the types of business in which these providers are involved. For example, a business organization may be of business type *Distributor* in a supply chain domain. For each business type, a set of useful e-services can be defined. Business organizations of the same business type may provide all or some of these e-services. To standardize the specification of an e-service, an *e-service template* can be jointly defined by those business organizations of the same business type. It consists of the following three general types of attributes:

- Input attributes, which specify the data needed as input to invoke an e-service.
- Output attributes, which specify the returned data of an e-service.
- Service attributes, which specify other properties of an e-service, such as the length of time the e-service takes, the cost for using the e-service, the quality of the e-service, etc.

An example of the e-service template for an e-service provided by business type *Distributor* is shown in Table 1.

Table 1 E-Service Template of e-service *Process Order* of *Distributor*

E-service	Description			
Process Order	E-Service Attributes		Name	Type
		Input Attributes	Product_Name Model_Name Quantity User_Info	String String Int UserInfo
		Output Attributes	Order_Status	Status
		Service Attributes	Duration	Time

All e-service templates are managed by the Broker Server [Hela01] and used by service providers to register their e-services and by process model designers to specify their service requests in a process model. A service provider registers its e-services with the Broker Server by first browsing and selecting the proper e-service template, which is displayed as a form to be filled by the service provider. During the registration, the service provider provides the broker with its general information such as its name, URL, telephone, email, etc. It then specifies which e-services it provides. For each e-service, the service provider can also specify some constraints on the input attributes and service attributes. By allowing constraints associated with service attributes to be explicitly specified, we will extend the e-service specification, UDDI [UDDI], to increase its expressive power. The constraints can either be attribute constraints, which specify the values that the individual input and service attributes can have, or inter-attribute constraints, which specify the interrelationship between the values of these attributes. These constraints restrict the kind of data that the requester of an e-service can provide when the e-service is invoked. For constraint specifications, we adopt the syntax and semantics of the Constraint-Based Requirement Specification Language used in [Huan99]. We shall call these constraints *e-service constraints.* For example, a distributor named *Worldwide* who provides the e-service named *Process Order* may specify the following constraint:

```
ATTRIBUTE_CONSTRAINT:
        Product_Name    String   ENUMERATION ["Computer"]
        Model_Name      String   ANY
        Quantity        Integer  RANGE[1-1000]

INTER_ATTRIBUTE_CONSTRAINT:
        Iac1  quantity > 500     implies     Duration>10
```

This constraint specification states that the e-service *Process Order* can only process the order of computer product with the quantity less than 1000, and if the quantity of the order is larger than 500, this e-service needs to take more than 10 time units. Iac1 is the name of the inter-attribute constraint.

For each e-service, the e-service attributes defined in the e-service template and the e-service constraints defined by the service provider together form the *e-service specification.* After registration, the general information of the service provider and the *e-service specification* of the e-service it provides are stored in a persistent store and managed by the Broker Server.

During process modeling, the *e-service requests* specified in a process model are defined in terms of the attributes given in their corresponding e-service templates. In addition to the values of the input attributes, the constraints on the service attributes can also be specified in an e-service request. We shall call the constraints in an e-service request *e-service request constraints* (in contrast to the *e-service constraints* discussed above). An example of an e-service request constraint is shown below.

```
ATTRIBUTE_CONSTRAINT:
        duration        int     [0 .. 10]        priority[1]
        cost            float   [0 .. 1000]      priority[2]
INTER_ATTRIBUTE_CONSTRAINT:
              Iac1   duration >4   implies   cost < 800
```

The above constraint specification states that the requester of this e-service request expects that the e-service should not take more than 10 time units, the cost of the e-service should not be more than $1,000, and if it takes more than 4 time units, then the cost must be less than $800.

An important function of the Broker Server is to do constraint-based brokering and service provider selection. To achieve this, the Broker Server would match an e-service request with e-service specifications given by service providers to identify the proper service providers for the request. The data provided for the input attributes of an e-service and the constraints specified in the request will have to match with (i.e., not in conflict with) the attribute constraints and inter-attribute constraints specified by a service provider. This function is used by the Workflow Engine to do *dynamic service binding* during the execution of a workflow instance; i.e., to bind e-service requests specified in the activities of a process model to the proper service providers.

4.2 Dynamic Workflow Model

WfMC's WPDL defines a well-accepted standard of workflow meta-model which can be used for the workflow vendors to exchange workflow models. We will use it as the basis of our dynamic workflow model (DWM). However, in order to support the dynamic nature of e-business, we made the following extensions and modifications to the underlying workflow meta-model of WPDL. These extensions and modifications make it easier to modify a process model both at build-time and run-time.

(1) Introduction of *Connector*

In WPDL, the specifications of the Join and Split constructs and their constraints (OR and XOR) are used to define the structural relationships and constraints among activities. In WPDL, they are defined as a part of the activity specifications. Thus, any change made to these constructs and constraints would entail change to activity specifications. We extract the specification of Join and Split constructs and their constraints (OR, AND, and XOR) from the activity definition and introduce a new modeling construct called *Connector*, which is used to specify the above aggregation properties. By separating activity specifications from the specifications of control information, any change made in one will not affect the other.

(2) Encapsulation of the activity definition

We extend the WPDL's activity definition by adding the specification of input parameters and output parameters. An activity can only reference the data passed by the input parameters. It exposes the result of operations (or tasks) specified in the activity only through the output parameters. By doing this, an activity definition is encapsulated. The activity definitions then can be reusable.

(3) Inclusion of explicit data flow specification

In a process model defined in WPDL, all activities can make reference to all the "workflow reference data" like global variables. The data flows are implicitly defined through the use of global variables. This makes the data flow relationships among activities unclear. In our model, we define data flows explicitly. We use inter-activity parameter mappings to define the data flows among the activities of a process model.

(4) Inclusion of e-service requests in activity definitions

In a process model defined for an organization, an activity specification can include manual or automated tasks performed within the enterprise. In a process model defined for an inter-organizational workflow, the activity specification should include e-service request(s) that can be serviced by different business organizations. As explained in Section 4.1, e-service requests in a process model are defined based on the corresponding e-service templates. E-service request constraints can be defined to restrict the selection of service providers to be bound to these requests.

An activity may contain several e-service requests. We shall assume that all these e-services are provided by the same business organization. Otherwise, the activity should have been split into two or more activities, each of which contains e-service requests that can be serviced by an organization. The *business type* of the business organization, whose e-services are requested in the activity, is also specified in an activity definition as the *performer* of the activity. The *performer selection constraint* is defined to further restrict the selection of a proper organization as the service provider of an e-service. There are four types of performer selection constraint: 1) CONSTANT, which restricts the service provider to a specific organization; 2) ANY, which specifies that the e-service request will be dynamically bound to a service provider determined at run-time by a Broker Server; 3) SAME_AS, which specifies that the service provider of this activity is the same as another specified activity, and 4) VARIABLE, which specifies that the service provider of this activity is computed by another activity.

(5) Introduction of events, triggers, and rules

Another important extension we make to the WPDL is the introduction of events, triggers, and rules in a process model specification. The activities inside the process model can post events. We distinguish the following three types of events:

- Before-Activity-Event: Before an activity is executed, the Workflow Engine that oversees the processing of a workflow instance would post a Before-Activity-Event.
- After-Activity-Event: After the processing of an activity, the Workflow Engine would post an After-Activity-Event.
- External events: An activity can also explicitly post events that have been defined and published by other users/organizations on the Internet. Posting an external event is regarded as an operation or task item in an activity.

We refer to Before-Activity-Event and After-Activity-Event as *workflow events* because they are treated as an integral part of a process model. Different from external events posted by activities, workflow events are automatically generated. That is, during business process modeling, a process model designer can specify whether an activity posts synchronous workflow events, asynchronous workflow events, or both. Event definitions are then automatically generated based on the name of the process model, the name of an activity, and the input data to the activity. The posting of an event can pass the input data to the rules that are triggered by the event.

By introducing these events, the execution of a workflow instance would post events to automatically trigger the processing of some business rules. These rules have the format Condition-Action-AlternativeAction. They may simply perform operations in addition to the task items specified in activities to enforce some business policies, regulations, or constraints. They may also modify the execution flow of the workflow instance (e.g., skip the next activity or branch to a specific activity in a process model). Different workflow users may "attach" different sets of business rules to the events of a process model when they enact the model. Thus, different workflow instances will trigger a different set of rules. In this way, a process model can be tailored to fit individual organizations' needs. The relationship between events and rules are specified by triggers.

By extending WPDL in the ways described above, we form our dynamic workflow model (DWM). In DWM, the modeling constructs used to define a process model include activities, transitions, connectors, subflows, data flows, events, rules and triggers. The graphical representation of a business process model is shown in Figure 3. The nodes in the graph can be activities, connectors, or subflows. The solid edges represent conditional transitions between activities, connectors, and subflows. The connectors and transitions together form the control flows, which represent the control dependencies among the activities and subflows in the process model. The data dependencies among the activities and subflows are captured by data flows. They can be either implicitly defined together with the transitions, or they can be explicitly defined. A thick solid line between activities/subflows in the graph represents an explicitly defined data flow. The ovals inside activities represent the events posted by the activities. The three types of events that can be posted by activities are Before-Activity-Event (BE), After-Activity-Event (AE), and External Event (EE). Business rules can be attached to these events by using trigger specifications (represented by dished lines).

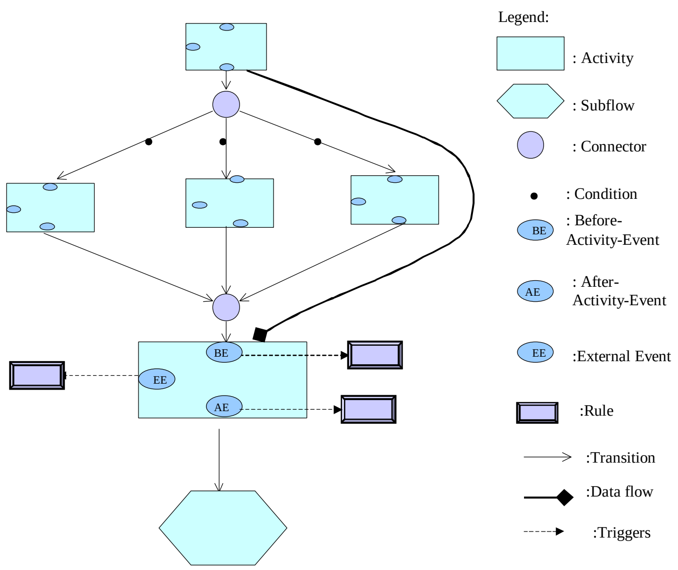


Figure 3 Business Process Model in DWM

4.3 Dynamic Properties Provided by DWM

The dynamic properties of DWM include four aspects:

- **Active**: Business events and business rules are integrated with business processes. A business process is active in the sense that its enactment may post synchronous and/or asynchronous events to trigger the processing of business rules. Synchronous events may trigger rules, which may dynamically alter the process model at run-time. These rules are defined either by the process model designer or the organization that initiates a workflow instance. Asynchronous events are notifications of the processing milestones of an enacted business process or exceptions. Interested organizations can subscribe to these events, and they can define business rules to react to these events. These business rules may enact other business processes.
- **Flexible**: The e-service requests specified inside a process model are defined according to standardized e-service templates. These e-service requests are bound to suitable service providers in a virtual enterprise during the enactment of the business process through a ***dynamic service binding*** mechanism. Thus, a process model defined in this way is flexible in the sense that the actual business organizations that take part in the business process are not determined until at run-time. Changes in the membership of the service providers of a particular service do not affect a business process model.
- **Adaptive**: The process models defined in DWM are adaptive since they can be easily modified to adapt to the changing business environment. Modifications to activity definitions or to the structural relationships and constraints of these activities can be more easily done because their specifications are separated. The workflow engine in our workflow management system provides APIs to do the run-time modifications of a process model, i.e., to delete/add transition, delete/add dataflow, replace an activity and/or modify a condition of a transition. These modifications can either be done by the business rules triggered by synchronous workflow events, or by the user who monitors the processing of a workflow instance.
- **Customizable**: Inter-enterprise process models are designed for conducting the business of a virtual enterprise. All the business organizations that participate in the virtual enterprise and have the right of access to a process model should be able to enact the model. In every enactment of a process model (i.e., each workflow instance), an organization may want to customize the process model to suit its

business policies, constraints or regulations. This can be done by defining its own set of business rules to add and/or modify the process model in all the enactments. These rules are triggered by the events posted by the process model. We shall call this type of customization the ***organizational customization***. Also, in different enactments of the same process model, an organization may want different rules to be triggered. This can also be done by defining different sets of rules for different enactments. We call this type of customization the ***instance customization***.

5. Sample Scenario: Order Processing in a Supply Chain Community

Suppose several organizations form a supply chain named *Supply_Chain_Community*. The organizations in this virtual enterprise are categorized into four different business types according to different roles they play: *Retailer, Distributor*, *Manufacturer*, and *Transportation Agency*.

A process model, *OrderProcessing,* defined in DWM can be used by distributors in the *Supply_Chain_Community* to process the orders issued by retailers, as illustrated in Figure 4. The *Distributor* gets an order from a retailer (Acticity A1), and checks the inventory to make sure that the quantity of the product in the inventory can satisfy the order (A2). In this scenario, we assume that the ordered quantity can be satisfied. The *Distributor* adjusts the inventory accordingly. It then acknowledges the order (A3) and asks the *Transportation Agency* to ship the product to the *Retailer* (A4). Note that the connector after Transition T3 has and AND constraint, specifying the following two activities can proceed in parallel. Transitions T6 and T7 also have an AND constraint, which specifies that both A3 and A4 have to be finished before proceeding to the End Activity.

In Figure 4, activities are represented by rectangles. The descriptions of the activities are inside the corresponding boxes. The business type information (i.e., performer) of each activity is indicated above the box. The performer selection constraint is enclosed within parentheses following the business type. For example, " ANY" following the business type *Transportation Agency* in Activity A4 means that the e-services requests in the activity can be serviced by any transportation agency, which can be bound to the service requests at run-time. The constraint "SameAs A1" in Activity A2 specifies that the performer of Activity A2 must the same as the performer of A1.

Three events are specified in the process model: a synchronous Before-Activity-Event in A1 and two asynchronous After-Activity-Events in A2 and A4, respectively. A Begin activity and an End activity are included in a process model to indicate the entry point and the exit point, respectively. To avoid cluttering the figure, we do not show the dataflow information and the detailed specification of the activities.

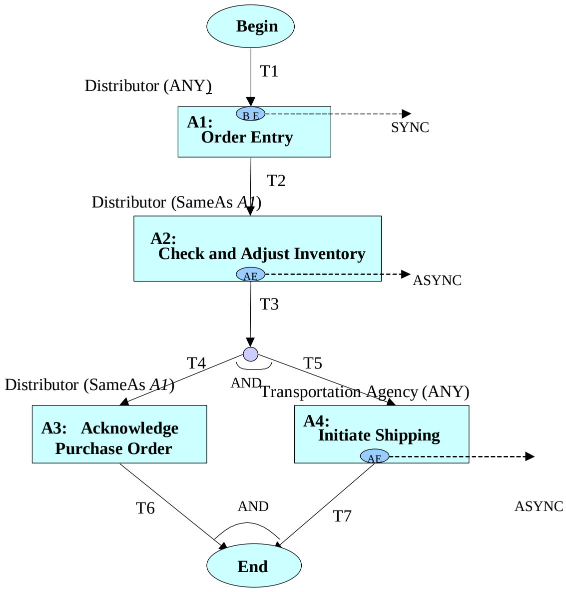


Figure 4 OrderProcessing Model for Distributors in the Supply_Chain_Community

This process model is replicated at all the ISEE Hubs of the organizations participating in the virtual enterprise. If an organization wants to make use of this model for its business process, it can customize the model by defining business rules, which reflect the local business policies, and by connecting them to the synchronous workflow events of the model by using trigger specifications. For example, the distributor *Worldwide* can define a business rule to check the credit history of the retailer who submitted the order, and attach it to the synchronous event that is posted before activity A1 (Order Entry). Organizations in the virtual enterprise can also subscribe to the asynchronous workflow events and attach business rules to it. For example, if the distributor *Worldwide* wants to check if the replenishment to the inventory is needed after adjusting the inventory, it can define a business rule to activate the replenishment process and attach it to the asynchronous event posted after activity A2.

6. Design and Implementation

This Section describes the design and implementation of the two main components in the Workflow Server: the Process Definition Tool and the Workflow Engine.

6.1 Process Definition Tool

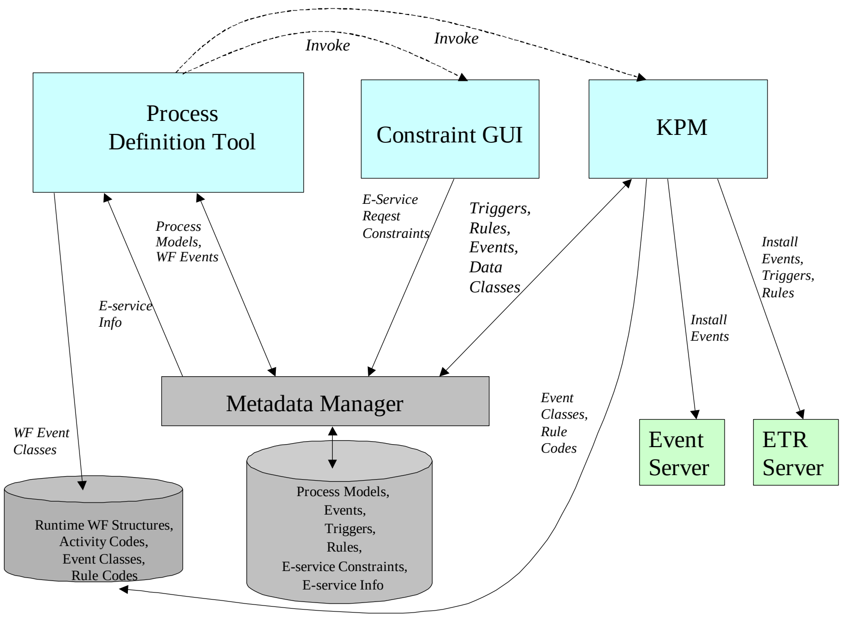


Figure 5 Build-time Environment of the DynamicWorkflow Management System

The Process Definition Tool is a user-friendly graphical editor for creating and customizing process models using DWM as the underlying model. The components of the Process Definition Tool are shown in Figure 5. It can be used to specify the diagram of a business process model, data objects manipulated by the business process, and the e-service requests details. The e-service requests are defined based on the e-service templates accessible from the Broker Server. For each e-service request, the process model designer can specify e-service constraints by invoking the Constraint Definition GUI. The Process Definition Tool also invokes a Knowledge Profile Manager (KPM), which provides the GUIs for defining data classes manipulated or transmitted by the activities, and business rules and triggers that link the rules to synchronous workflow events. A Metadata Manager is used to store the meta-information of process models, e-service request constraints, data classes, events, rules, and triggers. Based on the stored meta-information, a Code Generator generates the run-time workflow control structure and activity codes, which are used for the run-time execution of a workflow instance. The code generated for the defined events are installed in an Event Server, and those generated for the defined rules and triggers are installed in an Event-Trigger-Rule (or ETR) Sever. KPM, Event Server, ETR Server and Metadata Manager were developed for another project called Iknet [LEE00, SU00]. They are a part of an ISEE Hub and are used as the components of the dynamic workflow management system. The Process Definition Tool is implemented using Java beans technology.

6.2 Workflow Engine

The enactment of a business process is performed by the Workflow Engine, which forms the core of the run-time environment. The run-time interactions between the Workflow Engine and other components of the workflow management system are shown in Figure 6.

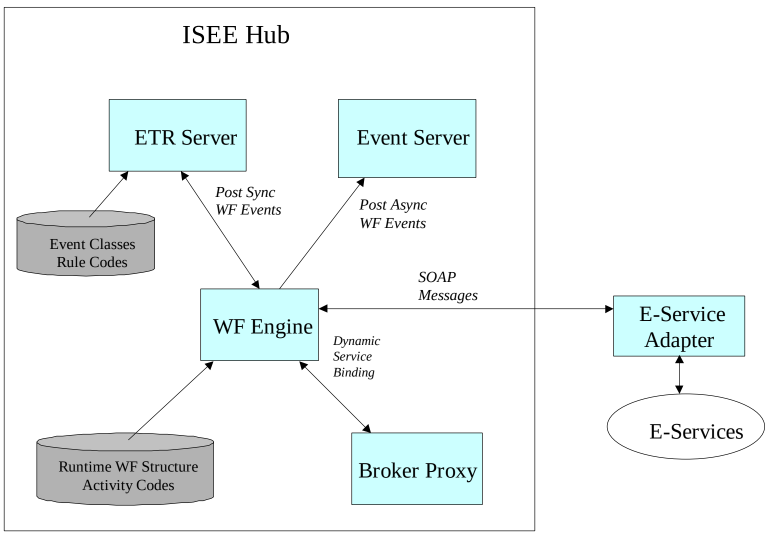


Figure 6 Run-time Interactions between the Workflow Engine and Other Servers

The Workflow Engine schedules the execution of a workflow instance based on the run-time workflow structures. When an activity is scheduled for execution, the corresponding activity code is dynamically loaded for execution. The Workflow Engine uses the brokering services provided by the Broker Server to bind the e-service requests in a process model to the suitable providers in a dynamic service binding process. To invoke an e-service, the Workflow Engine would send an XML message containing e-service request information to the e-service adapter at the site of the selected service provider.

Synchronous workflow events are posted directly to the ETR Server to trigger the associated business rules. Asynchronous workflow events are posted to the Event Server so that the Event Server can distribute them to their subscribers by event notifications.

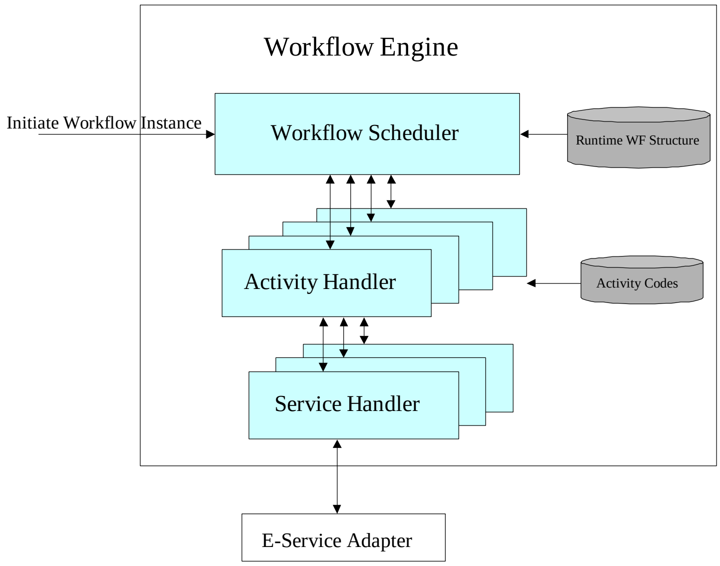


Figure 7 Workflow Engine Architecture

The architecture of the Workflow Engine is shown in Figure 7. The main components of the Workflow Engine are a Workflow Scheduler, an Activity Handler, and a Service Handler. The Workflow Scheduler schedules the transitions between activities, blocks, and sub-flows, and thus controls the execution of a workflow instance. It also handles data mappings between activities. When the Workflow Scheduler determines that an activity is ready to be executed, it delegates the execution of this activity to the Activity Handler. The Activity Handler is responsible for posting synchronous and asynchronous workflow events and dynamically loading and executing the activity code. The Service Handler is responsible for invoking the remote e-services. It sends the XML message containing the request data to the remote E-Service Adapter, according to the URL address obtained from the Broker Server during the dynamic service binding process, and receives the XML message containing the response data.

6.4 Run-time Modification to Process Model

The Workflow Engine uses run-time workflow structures to enforce inter-activity dependencies during activity scheduling. There are two kinds of inter-activity dependency: control dependency and data dependency. Control flow structures are used to capture the control dependencies, and data flow structures are used to capture the data dependencies.

The Workflow Engine provides APIs for the run-time modification of a business process model. The run-time modification is accomplished by modifying the run-time control structures and data flow structures of the process model.

- **Delete/Add a Transition:** The modification is accomplished by deleting/adding the corresponding control structures from/to the run-time workflow structures of the process model.
- **Delete/Add a Data flow**: The modification is accomplished by deleting/adding the corresponding data flow structure from/to the run-time workflow structures of the process model.
- **Replace an Activity**: Some business policies are embodied inside an activity body. When there is an unexpected change to these policies during the execution of a workflow instance, the running workflow instance may need to replace the activity with an alternative one. Because, in our implementation, the activity code is dynamically loaded when an activity is scheduled, we can achieve this by modifying the activity name in the old activity's control structure.
- **Modify Transitional Condition**: Similar to the modification of activities, the need for modifying transitional conditions also exists. In our implementation, the condition is evaluated in an interpretive way. We can just call the API provided by the Workflow Engine to modify a condition directly.

7. Summary of Our Research and Contributions

In summary, our research aims to develop a dynamic workflow management system for modeling and controlling the execution of inter-organizational business processes in a virtual enterprise environment. Our research efforts and contributions include:

- The design of a Dynamic Workflow Model (DWM) which extends the modeling capability of WfMC's WPDL. The extension includes adding the concepts of events, triggers, and rules into WPDL, introducing the connector construct, encapsulating activity definitions, and supporting e-service requests. The DWM, thus the dynamic workflow management system using DWM as the underlying model, is active, flexible, adaptive, and customizable.

- The introduction of a constraint-based, dynamic service binding mechanism for the enactment of business processes.
- The design and implementation of the dynamic workflow management system, which is a part of the ISEE information infrastructure for supporting e-business enterprises.
- The technique for run-time modifications of the run-time workflow structures to alter the course of executing workflow instances.

References

[Alon95] Alonso, G., Agrawal, D., El Abbadi, A., Mohan, A., Kamath, M., Guenthoer, R., "Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management," Proceedings of IFIP Working Conference on Information Systems Development for Decentralized Organizations, 1995.

[Alon99] Alonso, G., Fiedler, U., Hagen, C., Lazcano, A., Schuldt, H., Weiler, N., "WISE – Business to Business E-Commerce," Proceedings of 9th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises, Sydney, Australia, March 1999.

[Ceri97] Ceri, S., Grefen, P., Sanchez, G., "WIDE-A Distributed Architecture for Workflow Management," Proceedings of the 7th International Workshop on Research Issues in Data Engineering, Birmingham, UK, April 1997.

[Cros00] CrossFlow Consortium, "Flexible Change Control," CrossFlow Deliverable: D8.a, February 07, 2000, http://www.crossflow.org/.

[Ellis95] Ellis, C., Keddara, K., Rozenberg, G., "Dynamic Change within Workflow Systems," Proceedings of Conference on Organizational Computing Systems (COOCS), Milpitas, CA, 1995, 10-22.

[Geor95] Georgakopoulos, D., Hornick, M, Sheth, A., "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure," Distributed and Parallel Databases, 3, 119-153 (1995).

[Gepp98] Geppert, A., Tombros, D., "Event-based Distributed Workflow Execution with EVE", Proceedings of IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware '98), The Lake District, September 1998.

[Gref98] Grefen, P., Hoffner, Y., "CrossFlow - Cross-Organizational Workflow Support for Virtual Organizations," Proceedings of 9th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises, Sydney, Australia, March 1999.

[Hela01] Helal, A., Wang, M., Jagatheesan, A. and Krithivasan, R., "Brokering Based Self-Organizing E-Service Communities," Proceedings of the 5th International Symposium on Autonomous Decentralized Systems (ISADS) with an Emphasis on Electronic Commerce, March 26-28, 2001, Dallas, Texas.

[Huan99] Huang, C., "A Replicable Web-Based Automated Negotiation Server for Electronic Commerce Systems," Ph.D. Dissertation, Department of Computer and Information Science and Engineering, University of Florida, 1999, http://www.cise.ufl.edu/tech-reports/tech-reports/tr99-abstracts.shtml, TR 010.

[Kim00] Kim, Y., Kang, S.H., Kim, D., "WW-FLOW: Web-Based Workflow Management with Runtime Encapsulation," IEEE Internet Computing, May-June 2000.

[Klin98] Klingemann, J., Wäsch, J., Aberer, K., "Adaptive Outsourcing in Cross-Organizational Workflows," GMD Report, August 1998, http://www.gmd.de/publications/report/0030/.

[Lazc00] Lazcano, A., Alonso, G., Schuldt, H., Schuler, C., "The WISE Approach to Electronic Commerce," International Journal of Computer Systems Science & Engineering, special issue on Flexible Workflow Technology Driving the Networked Economy, Vol. 15, No. 5, 2000.

[Lee00] Lee, M., "Event and Rule Services for Achieving a Web-based Knowledge Network," Ph.D.Dissertation, Department of Computer and Information Science and Engineering, University of Florida, 2000, http://www.cise.ufl.edu/tech-reports/tech-reports/tr00-abstracts.shtml, TR 002.

[Mull99] Muller, R., Rahm, E., "Rule-based Dynamic Modification of Workflows in a Medical Domain," Proceedings of BTW99, Springer, Berlin, 1999, 429-448.

[Reic98] Reichert, M., Dadam, P., " Adept_flex-Supporting Dynamic Changes of Workflows Without Losing Control," Journal of Intelligent Information Systems, Special issue on Workflow and Process Management, Vol. 10, No. 2, 1998.

[Sheth97] Sheth, A., Kochut, K.J., " Workflow Applications to Research Agenda: Scalable and Dynamic Work Co-ordination and Collaborative Systems," in Advances in Workflow Management Systems and Interoperability, Eds.: Dogaç, A., Kalinichenko, L., Özsu, M. Tamer, Sheth, A., NATO Advanced Study Institute, Istanbul, Turkey, August 1997.

[Sheth99] Sheth, A., Aalst, W., Arpinar, I., "Process Driving the Networked Economy," IEEE Concurrency, Vol. 7, No. 3; JULY-SEPTEMBER 1999, pp. 18-31.

[Stri98] Stricker, C., Riboni, S., Kradolfer, M., Taylor, J., " Market-Based Workflow Management for Supply Chains of Services," Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, USA, 1998.

[Su00] Su, Stanley Y. W. and Lam, Herman, "IKnet: Scalable Infrastructure for Achieving Internet-based Knowledge Network," invited paper, Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, l'Aquila, Rome, Italy, July 31-Aug.6, 2000.

[UDDI] UDDI Technical White Paper, http://www.uddi.org.

[WfMC95] WfMC, The Workflow Reference Model, 1995, http:// www.wfmc.org.

[WfMC99] WfMC, "Interface1: Process Definition Interchange V 1.1 Final (WfMC-TC-1016-P)," 1999, http:// www.wfmc.org.