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, University of Florida, Gainesville, Florida 32611
{jmeng,su,hlam,helal}@cise.ufl.edu

Abstract

In the competitive global marketplace, business organizations often need to team up and operate as a virtual enterprise to achieve common business goals. Since the business environment of a virtual enterprise is highly dynamic, it is necessary to develop a workflow technology that is capable of handling dynamic workflows across enterprise boundaries. This paper describes a dynamic workflow model and a dynamic workflow management system for modeling and controlling the execution of inter-organizational business processes. The model extends the underlying model of WfMC’s WPDL by adding connectors, events, triggers and rules as its modeling constructs, encapsulating activity definitions, and allowing e-service requests as a part of the activity specification. The workflow management system makes use of an event and rule server to trigger business rules during the enactment of workflow processes to enforce business constraints and policies and/or to modify the process model at run-time. It also provides a mechanism to dynamically bind e-service requests to e-services.

1. Introduction

The advent of Internet, Web and distributed computing technologies has enabled business organizations to conduct business electronically: thus, the birth of e-business. E-business started with Electronic Data Interchange (EDI [1]) and evolved to business-to-consumer e-commerce, then to eMarketPlace portals. The current trend is toward collaborative e-business in which business organizations form virtual alliances under rapidly changing market conditions and make use of the best of their individual resources for achieving their common business goals. Business processes and application system services of the participating organizations of a virtual enterprise are important resources that need to be used to conduct a joint business. Workflow technology is an enabling technology for managing, coordinating and controlling the activities of a virtual enterprise.

The workflow of a virtual enterprise is an inter-organizational workflow since it is built on the workflows of participating organizations and makes use of the software services provided by them. Unlike a traditional workflow management system, which handles business processes within a single organization, an inter-organizational workflow management system must allow 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 of 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 business 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. Finally, 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 present a dynamic inter-organizational workflow solution to meet the above requirements. As an extension of the underlying model of Workflow Management Coalition’s Workflow Process Definition Language (WfMC’s WPDL [2]), we propose a dynamic workflow model (DWM), which enables the
specification of dynamic properties associated with a business process model in terms of events, triggers and rules. These new modeling constructs enable the integration of 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 other business processes and/or 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. The e-services are categorized according to different business types, for which standardized e-service templates are defined. Service providers register their e-services with a broker by using the templates. E-service requests are specified in the activity definitions of a process model. Their specifications are also based on the e-service templates. E-service requests are bound to the proper service providers at run-time by using the services of a broker to identify the suitable providers. By using the above dynamic binding approach, 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 specifications of process models. 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, the 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 Coalition has developed a framework for the establishment of workflow standards based on a Workflow Reference Model [3]. 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 (i.e., WPDL). It focuses on the specification of a process definition meta-model. This meta-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 of the 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. Most of the early works have been surveyed in [4]. In this section, we review some recent research efforts that are closely related to ours.

Events and rules have been used in several workflow projects. An example is the WIDE project [5]. 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. It also provides transactional support to workflow management. Events and rules are also used in the EvE project [6] 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 organizations of the 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 dynamic changes of process models used in the traditional workflow systems. The work reported in [7] 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 enable users to modify the structure of a running workflow, while maintaining its (structural) correctness and consistency. The work reported in [8] describes a rule-based approach for the detection of semantic exceptions and for dynamic workflow modifications, with the 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 work mentioned above, 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 [9] [10]. 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, which serves 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 [11]. Its goal is to develop and implement a mechanism for connecting WMIs and other WMS-like systems of different organizations in cross-organizational workflows and electronic commerce settings. CrossFlow defines a service-oriented model for cross-organizational workflows. In that model, a service specifies which part of a workflow it fulfills. For an external service, service selection at run-time is 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 [12].

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 system are defined and provided independent of business process models. E-service requests issued by these models are bound to e-services of service providers that are identified at run-time through a dynamic service binding process. Our system makes use of an Event-Trigger-Rule server to achieve more dynamic properties than the 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)). These servers provide ISEE-services to support the requirements of collaborative e-business. For example, the Event Server enables flexible and dynamic communication among loosely coupled systems that are distributed across organizations. The Event-Trigger-Rule Server provides timely and automated responses to events. Other available ISEE-servers for supporting collaborative e-business include an Automated Negotiation Service, a Constraint Satisfaction Processing Server, and a Cost Benefit Evaluation Server. Another important server provided by the ISEE infrastructure is the Dynamic Workflow Server. A dynamic workflow management system, which makes use of this server, will be described in the next section.
3.2 Architecture

The global architecture of the dynamic workflow management system is shown in Figure 2. The system consists of a network of replicated 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 for managing the e-service specifications that have been registered by participating business organizations, and for matching e-service requests against these specifications to select 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 management system](image)

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 using the constructs of a dynamic workflow model (DWM). An enactment of a process model is called a dynamic workflow model (DWM). An enactment of an inter-organizational process model may invoke manual and automated tasks that can be carried out by different participating business organizations. These “local” process models can be enacted by the 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 during the execution of an inter-organizational business process. An e-service adapter needs to be implemented and installed at each organization’s site as a wrapper of its e-services operations. The main function of the adapter is to map an e-service request to the API of an application that performs a specified operation. In our work, SOAP [13] messages are used to invoke e-service operations, which are mapped to local RMI calls to Java programs that perform the operations [14].

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 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 of defining e-services, it is useful to categorize and standardize e-services and their specifications by the types of business that these providers are conducting. 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 the business organizations of the same business type. An e-service template consists of one or more operations offered by the e-service. For each operation, there are three types of attributes:

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

An example of the e-service template for an e-service OrderProcessing provided by business type Distributor is shown in Table 1. It contains an operation Process Order.

<table>
<thead>
<tr>
<th>E-Service</th>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrderProcessing</td>
<td>Process Order</td>
<td></td>
</tr>
<tr>
<td>Input Attributes</td>
<td>productName String ENUMERATION [&quot;Computer&quot;]</td>
<td></td>
</tr>
<tr>
<td>Input Attributes</td>
<td>modelName String ANY</td>
<td></td>
</tr>
<tr>
<td>Input Attributes</td>
<td>quantity Integer RANGE[1-1000]</td>
<td></td>
</tr>
<tr>
<td>Output Attributes</td>
<td>status Status</td>
<td></td>
</tr>
<tr>
<td>Service Attributes</td>
<td>duration Time</td>
<td></td>
</tr>
</tbody>
</table>

All e-service templates are managed by the Broker Server [15] 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 needs to specify the e-service binding description, which contains the location of the service implementation and details on the protocol and the port to be used to access the server that hosts the e-service. The service provider can also specify some constraints on service attributes of the operations. By allowing constraints associated with service attributes to be explicitly specified, we have extended the e-service specification, UDDI [16], to increase its expressive power. The constraints can be either attribute constraints, which specify the values that 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 [17]. We shall call these constraints e-service constraints. For example, a distributor named Worldwide who provides the e-service named OrderProcessing may specify the following constraint on the operation Process Order:

ATTRIBUTE_CONSTRAINT:

- productName String ENUMERATION ["Computer"]
- modelName String ANY
- quantity Integer RANGE[1-1000]

INTER_ATTRIBUTE_CONSTRAINT:

- Iac1 quantity > 500 implies duration>10
This constraint states that the e-service Process Order can only process the order of a computer product with the quantity in the range of 1 to 1000, and if the quantity of the order is larger than 500, this e-service operation 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 its e-service specification 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 constraints (in contrast to the e-service constraints discussed above).

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 has a well-accepted, standard workflow meta-model. We 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 the process models defined in DWM, both at the 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. They are a part of the activity specifications. Thus, any change made to these constructs and constraints would entail change to the activity specifications. We extract the specifications of Join and Split constructs and their constraints (OR, AND, and XOR) from the activity definition and introduce a new modeling construct called Connector 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. The data flows may or may not follow the paths of the control flows.

(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 all participating business organizations. As explained in Section 4.1, e-service requests in a process model are defined based on their corresponding e-service templates. E-service request constraints can be defined to restrict the selection of service providers to be bound to these requests.

(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 and/or asynchronous workflow events. 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 different sets of rules. In this way, a process model can be tailored to fit individual organizations’ needs. Rules can also be dynamically attached to the events posted by a running workflow instance to handle situations that were not foreseen when the instance was initiated. The relationships between events and rules are specified by triggers.

By extending WPDL in the ways described above, we construct our dynamic workflow model (DWM). In DWM, the modeling constructs used to define a process model include activity, transition, connector, subflow, data flow, event, rule and trigger. 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, or subflows. The connectors and transitions together specify the control flow. 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 ending in a diamond shape 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:
(1) 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 define business rules to react to these events. These business rules may enact other business processes.

(2) 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 the 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 will not entail changes to a business process model.

(3) 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 the 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.

(4) 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. The set of rules would apply to all the enactments of 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 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 (Activity 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 an AND constraint, specifying that 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: asynchronous 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.

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

![Figure 4. OrderProcessing process model for distributors in the Supply_Chain_Community](image-url)
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. Implementation

We have designed and implemented a prototype of the dynamic workflow management system. The two main components of the Workflow Server are the Process Definition Tool and the Workflow Engine. The Process Definition Tool is a user-friendly graphical editor that can be used to specify the diagram of an inter-organizational business process model and details of e-service requests. It also invokes a Knowledge Profile Manager (KPM) GUI, which was developed for another project called IKnet [18, 19] to define business events, rules and triggers [20], and invokes the Constraint Definition GUI [17] to define the constraints associated with the e-service requests.

The enactment of a business process is performed by the Workflow Engine, which forms the core of the run-time environment. 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 a SOAP message containing e-service request information to the E-service Adapter at the site of the selected service provider. The E-Service Adapter parses the SOAP request message received from the Workflow Engine and invokes the implementation of the requested operation of the e-service. After the invocation is complete, the E-Service Adapter composes the SOAP response message and gives the result to the Workflow Engine.

The Workflow Engine posts synchronous workflow events 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 notify their subscribers at various ISEE Hubs by an event notification mechanism. The Event Server and the ETR Server were also developed for the IKnet project.

The run-time interactions between the Workflow Engine and other components of the workflow management system are shown in Figure 5.

![Figure 5. Run-time interactions between the Workflow Engine and other servers](image)

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.

The control flow structure of a process model is used to capture the control dependencies, and the data flow structure is 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 workflow structure of the process model. These APIs support functions like deleting/adding a transition, deleting/adding a data flow, replacing an activity, and modifying a transitional condition.

7. Summary

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-services. The DWM and, thus, the dynamic workflow management system using DWM are 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 introduction of the technique for run-time modifications of workflow structures to alter the course of executing workflow instances.

8. References


