View-based Contracts in an E-service Cross-Organizational Workflow Environment

Eleanna Kafeza[1], Dickson K.W. Chiu[1] AND Irene Kafeza[2]

[1]Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[2]Department of Law, City University of Hong Kong, Kowloon, Hong Kong
kafeza@cs.ust.hk, kwchiu@ieee.org, irene.k@plink.cityu.edu.hk

Abstract. In an e-service environment, workflow involves not only a single organization but also a number of business partners. Workflow inter-operability is therefore an important issue for workflow enactment in such an environment. In this paper, we introduce a novel concept of workflow views as a fundamental support for E-service workflow inter-operability and for controlled visibility by external parties. Furthermore, we develop a contract model based on workflow views and demonstrate how management of contracts can be facilitated, with an Internet start-up E-service inter-organization workflow example.

1 Introduction

Workflow is the computerized facilitation or automation of a business process. A business process is a set of one or more linked procedures or activities, which collectively realize a business objective or policy goal. Workflow Management Systems (WFMSs) can assist in specification, decomposition, coordination, scheduling, execution, and monitoring of workflows. In this paper, we use the term workflow to refer to this more general notion of process management. Besides streamlining and improving routine business processes, WFMSs can help in documenting and reflecting upon business processes. Traditional WFMSs often can only coordinate workflows and their enacting agents (often limited to software processes) within a single organization.

The Internet has recently become a global common platform where organizations and individuals communicate among each other to carry out various commercial activities and to provide value-added services. E-service refers to services provided via the Internet. Therefore, there is an impending need for supporting cross-organizational workflows to these activities, especially because many organizations may have already been employing some kind of workflow technologies. Advanced WFMSs are now web-enabled and recent researchers in workflow technologies are exploring cross-organizational workflows to model these activities. In addition, advanced WFMSs can provide various services such as coordination, interfacing, process repository, process (workflow) adaptation and evolution, match-making, exception handling, data and rule bases, etc, with many opportunities for reuse.

We have some preliminary work [4] to demonstrate the feasibility of modeling and enacting composite E-service as workflow extensions, so that we can build E-service agents (i.e., a system that provides E-service as delegated by users), and the system for supporting them quickly, with all the desirable features provided by the underlying WFMS. Furthermore, we have proposed a novel concept of workflow view in [5] for supply-chain management in a cross-organizational workflow environment. As follow-up work, we detail in this paper how composite E-services can be modeled as cross-organization workflow, with respect to our E-ADOME workflow engine extended with various agent interface. In addition, we apply a promising novel approach using workflow views for contract modeling and enforcement. Views help balance trust and security, i.e., only information necessary for the process enactment, enforcement and monitoring of the contract is made available to both parties, in a fully control and understandable manner. Moreover, each party only needs minor or even no modification to its own workflow, but can successfully arrive at a commonly agreed and interoperable interface. This kind of adaptation (fully support by E-ADOME [4,7]) is only required upon their first contract, and reusable subsequently, unless their workflows are changed drastically. Because an organization is probably making lots of contracts with many other different organizations, different views of a workflow can be presented to different organizations according to different requirements. Thus, inter-organization workflows can be developed fast and managed adequately, together with e-contracts, since the E-service arena is very competitive.

The contribution and coverage of this paper are as follows: (i) a cross-organization workflow viewpoint of a composite E-service with a novel concept of workflow views, (ii) a contract model based on workflow view, (iii) illustrates how workflow views facilitate e-contract management, such as process adaptation for interoperability and contract enforcement.

The rest of our paper is organized as follows. Section 2 presents a motivating example to illustrate a novel concept of workflow views in an E-service cross-organizational workflow environment. Section 3 presents our view-based model for e-contracts. Section 4 illustrates how workflow views facilitate e-contract management, and how two organizations can arrive at an e-contract with verification. Section 6 compares related work. Finally, we conclude the paper with our plans for further research in Section 6.

2 Workflow Views

In a B-to-B e-commerce environment, a business process usually involves many participating organizations, i.e., such a business process involves several inter-operating and interacting workflows from different organizations. This is known as cross-organizational workflow. To support workflow inter-operability, one of the basic requirements is a mechanism to let authorized external parties access and make use of only the related and relevant parts of a workflow, while maintaining the privacy of other unnecessary/unauthorized information. Motivated by views in federated object databases, we propose the use of workflow views as a fundamental mechanism for cross-organization workflow interaction. A workflow view can be either regarded as a structurally correct subset of a workflow definition (as in [17]) or a structurally correct composition of workflow definitions.

Workflow views are also useful in providing access to business processes for external customers or users, including B-to-C e-commerce and e-service. For example, external customers or users may want to check the progress or intermediate results of the business processes that they are participating. They may be required to provide additional information or make decisions during business processes. Even within an organization, workflow views are useful for security applications, such as to restrict accesses (like the use of views in databases).

We propose the use the concept of workflow views (which is detailed in the next section) to help advanced interactions among WFMSs and allow them to inter-operate in a white box mode (i.e., they can access some internal information of each other). In particular, we allow execution of another workflow in the same E-ADOME system to handle a task, as a form of dynamic reuse. For example, the task "Detailed Homepage Work" may be contracted out to an outside company as a workflow in another system. Alternatively, the task may be assigned to an in-house team as an extra workflow in the same E-ADOME system (cf. [4]).

Since ADOME-WFMS is event-driven, events and messages from other WFMS are intercepted by the E-ADOME layer and then presented to the ADOME-WFMS. As presented in the example in the following subsection, an event from a workflow of another organization (e.g., an inquiry from a customer) can trigger the start of a workflow in the local ADOME-WFMS, or used for synchronization purposes (e.g., deposit payment triggers ordering of a leased line). Similarly, an event from the local ADOME-WFMS (e.g., an inquiry to a supply) can trigger the start of a workflow in another organization, or used for synchronizing tasks in another organization (e.g., an order triggers delivery from another organization). As such, cross-organization workflow interactions can be facilitated.

2.1 Internet Startup E-Service: Cross-organization Workflow Viewpoint

In [4], we presented an E-service workflow example based on the Internet Startup Service of Dickson Computer Systems. This workflow actually involves many parties: end-users who need the service, Dickson Computer Systems as a value-added service-provider, and the vendors / basic-service providers of Dickson Computer Systems. In order to explore more detailed interactions among these parties, we proceed to present a cross-organizational workflow view among these parties. We present a multi-party example to illustrate E-service provision chains. This is a novel approach and a significant step forward from our previous work [4] in E-services.

In order to out-source a leased-line based Internet startup project, the end-user undergoes a requisition workflow (cf. Fig. 1(a)). First, quotation inquiries are sent to a number of E-service providers. The received quotations with service details are evaluated. An order form is sent to (filled in through a web interface) the selected E-service provider, say Dickson Computer Systems, with deposit payment. The leased-line and the web server are then installed. The end-user then participates in negotiation and approval of the detail homepage design work and eventually accepts the finished web site. Finally, the balance payment is arranged for.

An E-service provider's workflow starts when an enquiry is received. Fig. 1(b) depicts a workflow based on the Internet startup service Dickson Computer Systems in [4]. The additional required steps are to compile a quotation from update inquiries from its vendors the prices of a server PC and a leased line. It should be noted that multiple PC vendors and multiple leased-line providers might be contacted for selection. After order confirmation, the more accurate action upon the exception "domain name in use by others" is to inform the end user for immediate amendment. The order of a leased line is only triggered until the deposit payment, because leased-line installation charges are not refundable even upon order cancellation, while PC servers are most probably reusable for other customers. Furthermore, there are probably long lasting interactions in the step "detailed homepage work" for design negotiations. Finally, after receiving the payment, the workflow also ends.

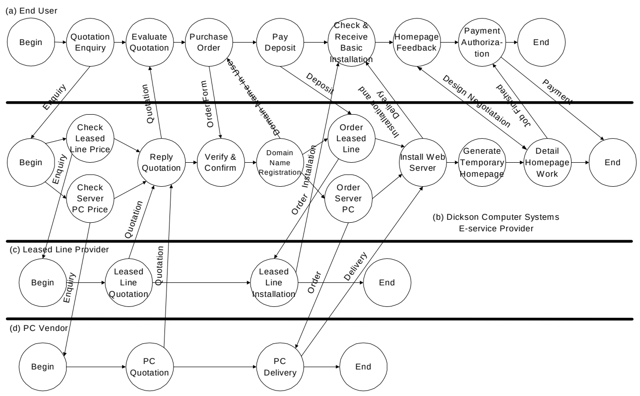

Fig. 1. Cross-Organiztional E-service Workflow

A leased-line provider's and a PC vendor's workflow also start when an enquiry is received. Assuming this is the end of the E-service provision chain, the vendors have all necessary information to quote the price. B-to-B orders on standard service packages are usually performed together with payment arrangements. It should be noted that the lease-line provider directly installs the lease line at the end-user's site and therefore directly triggers their "check and receive basic installation" step, without the need to go through Dickson Computer System. These two workflows end after the leased line is installed and after the PC is delivered, respectively.

In this example, The workflow view of the *end-user* presented to the system integrators has the following requirements: (i) The end-user's company profile and other background information are made available on request so that the system integrators can design more personalized proposals. (ii) Changes in delivery requirement, lease-line installation date, or payment arrangement should notify the E-service provider. (iii) In particular, the enquiry process is concealed so that the E-service provider can bid fairly and independently.

The workflow view of the *Dickson Computer Systems* presented to the *end-user* has the following requirements: (i) The company profile of Dickson Computer Systems and specifications service packages are made available on request so that the end-user can evaluate our proposals and quotations more accurately. (ii) Changes in any delivery schedule (hardware or services) should be notified to the end-user. (iii) The progress "Internet Startup Service" and other sub-workflows are available to the end-user so that the user can further monitor the progress of the job and estimate the delivery date. (iv) However, some trade secrets, such as the source and price of services and products, are not presented to the end-user to prevent them from purchasing directly from the vendors. (v) Updated quotation (price) is sent to the end-user upon a significant aggregated price change in hardware or service items with event-triggering mechanism during the evaluation process of the end-user.

The workflow views of the *product or service vendors* (e.g. PC, leased-line) presented to Dickson Computer Systems should have the following requirements: (i) Price for the services or products is updated with event-triggering mechanism. (ii) Technical specifications and related information for the services and products are made available upon request. (iii) Updates in software drivers and service configurations should notify Dickson Computer Systems using event-triggering mechanism (which in turn can notify the end-users). (iv) Changes in lead-time should also be notified. (v) Dickson Computer Systems can monitor the progress of leased line vendor in arranging their installation and web-page designer in the contract-out work so that this information is indirectly available to the customer.

2.2 A Model for Workflow Views

Formally, based on Workflow Management Coalition (WfMC) workflow definitions [21], a workflow is described by $W = (T, A, J, F, X, V, E)$ where T is the set of tasks, $A \in T \times T$ is the set of arcs in the transition graph, J is a Boolean function determines whether there is a join immediately before every task in T, F is a Boolean function determines whether a fork occurs immediately after every task in T, X is the condition function associating every element of A a condition, V is the set of variables, and E is the set of events[1].

A *workflow view* is a structurally correct subset of a workflow definition. Formally, a workflow restriction view $W' = (T', A', J', F', X', V', E')$ is based on $W = (T, A, J, F, X, V, E)$ such that $T' \subseteq T$, $J' \subseteq J$, $F' \subseteq F$, $X' \subseteq X$, $V' \subseteq V$, $E' \subseteq E$, and $\forall (a, b) \in A'$, $\exists n$, $\forall i$, $1 \le i \le n$, $(t_i, t_{i+1}) \in A$ where $t_1=a$, $t_n=b$ (the transitions in the view W' is based on any valid paths in W).

The components of a workflow include the process flow graph, input/output parameters, objects, rules, events, exceptions and exception handlers associated with the workflow. Thus a view for a workflow instance also contains these components. Though every component is modeled as objects in most advanced object WFMS, we discuss them separately because each of them has different semantics. Fig. 2 depicts a simple workflow view definition language.

---

[1] Rules are modeled as objects; and exceptions are events [6,7].

```
view v of workflow w begin
   {process p1 view v1 ...}
   {process p2 renames p3 ...}
   {transistion t renames p4 to p5 ...}
   {object o1(=expression1), o2(=expression2)...
         (write) (input) (output) ...)
   (attribute a1,a2,...,an write | read | denied ...}
   {event e1=expression1, e2=expression2, ...}
   {exception e1=expression1, e2=expression2, ...}
   {rule r1=expression1, r2=expression2, ....}
end
```

Fig. 2. Workflow View Definition Language

Process Flow Graph - Most contemporary WFMSs use a hierarchical composition approach, i.e. a process (workflow) is composed of sub-processes and so on down to leaf-nodes of atomic tasks. This provides a good granularity for providing views of the process flow graph. If a workflow view is to be made available, a fundamental provision is the topmost level process flow graph. However, the detail composition of individual sub-process may be concealed. Thus a process in the flow graph can be presented in one of the following ways. A white-box sub-process is specified with a sub-workflow view by a statement "process p1 view v1", i.e., the details of the sub-process is further visible and subject to the restriction of a sub-workflow view. A black-box sub-process (e.g., "Quotation Enquiry" in Fig. 1) is limited from further details of its further internal composition. Unless a view is specified for the sub-process, it is a black box. A gray box where some sub-processes are visible while other sub-processes are concealed (e.g., the whole end-user procurement process). Furthermore, since the name of a sub-process or a transition label may reveal some information, it can be renamed with a rename statement. The statement "process p2 rename p3" renames a process p3 to p2 while the statement "transition t renames p4 to p5" renames the transition from process p4 to process p5 as t.

Objects associated with a workflow instance - An object associated with a workflow instance need not be presented completely in a workflow view. Some attributes can be hidden from the view, some can be read only, some are presented with write access, while composite attributes can further be composed of attributes of different access. Moreover, derived objects specified with object-SQL can be presented in a view. Input / output parameters are also objects specified for the interaction of the user. These parameters are actively received from or presented to the user upon interaction or certain events, with other regular objects in a workflow view are available only upon user's request. The "object" statement in Fig. 4 presents an object in a view. The optional expression is used to specify a derived object. The write option grants write access to the view user. The output option specifies the content of the object to be actively sent to the user. The input option specifies the object to be updated from the user. When the access of some attributes of an object are different from the default read or write access specified by the "object" statement, it can be overridden by the "attributes" statement, where explicit read, write or denied access can be specified.

Events and exceptions - When events and exceptions are presented in a view, a mechanism, such as a corresponding message, should notify the view user upon their occurrences. This is particularly useful in providing cross-organizational process synchronization and constraint enforcement. Events and exceptions are specified with "event" and "exception" statements, respectively. In addition, the view provider should support user-specified events based on all their accessible objects and process states. In this way, the user need not poll on their interested objects and thus increase the efficiency. For example, the end-user may specify that changes in the delivery date be an event so that the user can be notified when the delivery is earlier or later then expected.

Rules and exception handlers - Rules are presented in a view so that a user can be aware of some of the actions taken by the provider upon certain events or exceptions. This is useful because the process flow graph cannot specify workflow actions that are taken in an asynchronous or event-driven manner. Some of these actions are exception handlers in the view provider. In this way, the user can avoid duplicating some error handling procedures if the errors have already been taken care of. Rules are specified with the "rule" statement, where they can be specified in (Event, Condition, Action) form or any composition of existing rules. In addition, constraints can be specified in the form of rules. Especially, rules can be used as integrity or semantic constraints in views. We are investigating in this direction and further details are beyond the scope of this paper.

3 An E-Contract Model Based on Workflow Views

An e-contract is an abstract of an agreement between two parties. Every contract has some basic information to be captured by an information system. In our e-contract model a *contract description D* is a set of attributes whose values describe the necessarily information in order to form a contract. Example attributes of the describe list are: *D*={*Accept*, *Offer*, *Goal*, *Schedule*, *Payment*, *Documents*, *QoS*, *Exception_Rules*, *Commit*, …} where *Offer* and *Accept* denotes the organizations that offers and accepts the contract respectively; *Goal* is the objective why the contract is formed; *Schedule* is a set of dates and items to be carried out, including that the contract starts, finish, and any milestone of progress; *Payment* is a set of rules and values for payment; *Documents* is a set of documents that have to be available to both parties before forming the contract; *QoS* is a set of attributes for the required quality of service; *Exception_rules* is set of event-condition-action (ECA) rules to specify anticipated exceptions and their consequences; and *Commit* denotes whether the parties are committed to the contract or not. Since different countries and organizations may pose different requirements for creating a contract, and because of domain-specific requirements, there might be other attributes depending on the case as well.

When forming a contract, besides the description part, the two parties have to agree on the task assignment. For example, in Fig. 1 we have a contract between Dickson Computer Systems and the leased line provider. Each party has its own internal workflow. In order to cooperate, each party must be able to view a subset of the workflow of the other party that will specify the tasks that is obliged to perform. The issue is that in every contract we have to balance two concepts: trust and security. When two parties are forming a contract we assume that there is trust between them and that information necessary for the specification, enforcement and monitoring of the contract is available to both parties. At the same time, for security reasons no party wants to reveal more than it is necessary to the other party. In our e-contract model, the balance is achieved though a workflow view mechanism. Each party specifies a view of its internal workflow that is accessible to the other party. For example, the end-user specifies at the view that the task *evaluate quotation* becomes visible to the Dickson Computer Systems. At the same time details (i.e. the sequence of tasks) that describe how the quotation is evaluated are not disclosed since the user does not want the other party to know the internal evaluation procedure.

Although we may assume a mechanism that enforces the flow of control in each party's workflow, the control flow has to be augmented with inter-organizational communications in order to support the specific contract. These communications are useful for information exchange, control exchange, synchronization, and exception handling.

In our e-contract model, cross-organizational control flow information is specified within *communicating tasks* and their associated *communication links*. In each view, there are some tasks, called *communicating tasks*, through which two parties communicate. For example in Fig. 1, the *Payment authorization* of the End-User's workflow has to interact with the *End* node of Dickson Computer Systems and send the *Payment*. It has also to interact with the *Detail Homepage work* and wait for the *Job finish* message. The communicating tasks of the views of the two parties exchange messages through cross-organizational communication links. Each communicating task receives and sends a set of messages.

When specifying a contract, the order in which these messages occur is crucial. For example, the *Payment* should be sent only after the *job finish* is received. Therefore, with every communicating task we associate a partial order on the messages that has to send/receive. Moreover, we say that two messages are related through a "*strong*" order if and only if a reverse order execution has as a result the breach of contract.

Definition: Let V_i and V_j be the two views of two parties that have formed a contract. We call tasks of the view V_i and V_j that has interaction *communicating tasks*, and we associate with these tasks a set of messages M, and a partial order on M $<$, that imposes an order on the messages.

For example, *pay deposit* is a communicating task of the end-user, while *order leased line* is a communicating task of Dickson Computer Systems. When the *pay deposit* task is executed, a deposit should be paid to Dickson Computer Systems. Therefore, we define a communication link as an arc from the *pay deposit* task to the *order leased line* task. The direction of the communication link specifies which party is responsible for initiating the communication. We call a task responsible for initiating the communication a *performance* task. We also associate the message as a label of the communication link.

There are two different types of communication links, *simple* and *obligatory*. When forming a contract the parties have to specify which cases give ground to breach of the contract, i.e., where one party does not *perform* as it should. The *performance* in our E-contract model contract is represented by the communication links. If a communication link is characterized as *obligatory*, but the party responsible for the communication fails to do so, then the other party has the right to terminate the contract or take appropriate measurements. For example, if the user in the (*pay deposit*, *order leased line*) communication link does not pay the deposit within a given deadline, Dickson Computer Systems has the right to terminate the contract.

It should be noted that the contract is in a negotiation state and does not formally starts before the first performance link, i.e., in this example before the order form is sent. We define the communication part of the contract that spans between two different organizations as a bipartite graph and completes our definition for a e-contract as follows:

Definition: Let $t_i \in V_1$ and $t_j \in V_2$ be two communicating tasks from two the workflow views V_1 and V_2. We define a *communication link* from t_i to t_j as a tuple (t_i, t_j, M, *type*) where t_i is the task to initiate the interaction between the two workflows by sending a message to t_j, M is the message specification ant *type* specifies whether this is a *simple* or *obligatory* link.

Definition: Let V_1 and V_2 be the two workflow views that participate in a contract and T_1 and T_2 the set of communicating tasks respectively. A *cross-organizational communications graph E* is a bipartite graph between T_1 and T_2, where each edge of the graph is a communication link.

Main Definition: An *e-contract* is a tuple (V_1, V_2, E, D) where V_1 and V_2 are workflow views that participate in the contract, E is a *cross-organizational communications graph* between V_1, V_2, and D is a *contract description* of the e-contract based on V_1, V_2.

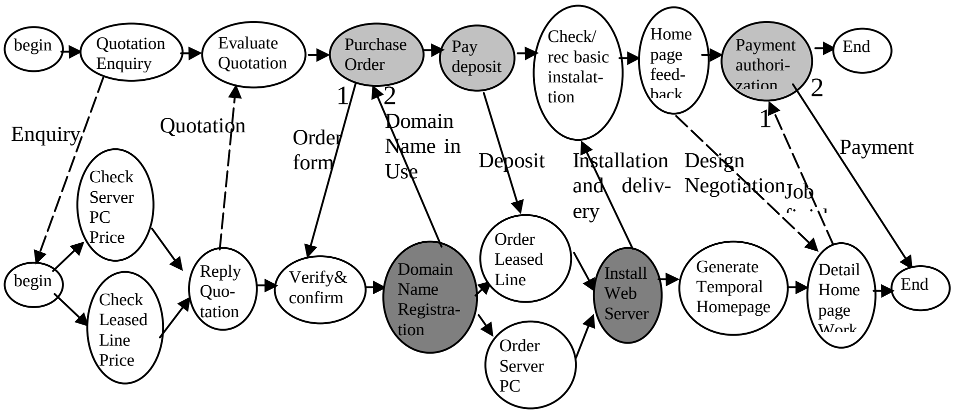

Fig. 3. A E-contract Communication Graph

In our graphical representation as depicted in Fig. 3, *simple* links are represented in dashed lines, while the partial order $<$ on the messages of each communicating task is represented by number labels at the end/start of each arc. We extract the messages associated with every communicating task by the labels of the bipartite graph. A performance task has a *send* message while the corresponding task *receives* message. We use light/dark gray color to represent the communicating tasks of the End-User/Dickson Computer System workflow views respectively, which are *performance tasks* and have *obligatory links*. The *Quotation Enquiry* task of the End-User is *a* *performance task*, i.e., it has to initiate an action - to communicate with the *Begin* node of Dickson Computer Systems by sending an *"Enquiry"* message. If this message is not sent, Dickson Computer Systems can still send the quotation. However, when the Purchase Order communicating task does not send a specific order form within a deadline, Dickson Computer Systems can assume that there is no obligation to perform subsequent tasks.

4 Managing E-contracts

In the previous section, we have described our e-contract model based on workflow views. In this section, we illustrate the processes related to e-contract management in a cross-organizational E-service workflow environment.

4.1 Workflow / View Adaptation during Negotiation

As long as there is no standardized workflow specification at an application level for each trade or service business, we perceive that workflow adaptation is a hard and tedious problem, which must be adequately addressed. When two organizations are interested in making an e-contract for a certain E-service, they exchange an initial workflow view of each other, to disclose their company profiles, and to inform the other party procedures involved in their organization, such as details of service packages of the service provider and the procurement procedure of the end-user. These views contain also the information and coordination requirements of both parties. However, these requirements often vary in different organization, i.e., workflows from different organizations may often have mismatches. The use of workflow views can now offer another advantage of shielding their underlying workflows from the necessary modifications. The following different levels of workflow adaptation may be required for interoperations of different organizations:

1. Workflow views can be modified to accommodate for interface mismatch and minor procedural differences without the need to modify the internal workflow.
2. Internal workflows need minor adaptation to accommodate for missing procedures (e.g., some companies usually do not pay deposit, therefore they need to add this task) and other minor logistic difference. This adaptation can be permanent if the organization believes it is useful for improving the business process (in dealing with other companies or other favorable reasons). This is known as *workflow evolution* [7]. Alternatively, the adaptation can be just a *deviation*, which is only employed in dealing with this particular business partner.
3. Because there may be major difference in workflows of the two parties, one or both of them decide to re-compose their workflows to accommodate for the cooperation. This case may be common, but few are willing to so. Alternatively, especially if the business relationship is not a long term one, one of the two parties may choose to fall back to a manual mode of cooperative (semi-manual) work-around. Since E-ADOME supports interfacing with human users through a web-based interface, for example, the end-user may designate a staff member to enter the order form

manually through the web-page of Dickson Computer System, and subsequent interaction are done through email, ICQ alerts, and further customized generated web pages.

Because an organization is probably making lots of contracts with many other different organizations, different views of a workflow can be presented to different organizations according to different requirements. In addition, workflow adaptations, which are sometimes required, are also well supported in E-ADOME. Thus, inter-organization workflows can be developed fast and managed adequately, together with e-contracts, since the E-service arena is very competitive. Otherwise, effective manual interaction through customized web pages is also supported by E-ADOME. However, a methodology for negotiation and workflow adaptation is beyond the scope of this paper.

4.2 Defining a E-Contract

After two parties have decided to make a contract, they have to arrive at an e-contract, which specifies the detail. In this section we present a methodology to define and verify a contract in accordance to our e-contract specification. When two parties want to form a contract, first they have to decide on the contract description set *D*, like the following example based on Fig. 3:

```
Create Description D
Accept: User
Offer: Dickson Computer Systems
Goal: Internet Startup Service
Schedule: {Start: June 30, 2001,
           Lease line installation: July 14, 2001,
           Server installation: July 16, 2000,
           ...,
      Finish: July 30 2001}
Payment:  {Before June 30, 2001: $1000 (Deposit),
           ...,
      With 14 days after Finish: Balance }
QoS:  Certified_Professions;
Exception_Rules:  {Schedule_delay <=7 days, do_nothing,
   Schedule_delay > 30 days : ...
   Leased_line.not_installable : ...}
Documents: Enquiry, Company Profiles, Order Form, Quotation
Commit: Yes
...
```

The description is the proof that both parties have agreed on the formation of a contract, and the e-contract model depicts the details. Then, each party has to present the view as specified in the e-contract model, in order to allow access to workflows of each other, and to incorporate the contract requirements on the data and control flow. Moreover, each party has to augment the communicating tasks with necessary communication links. In this example, after the End-User has executed the *Set_Communications* algorithm, it augments the communicating tasks with communication links as follows:

node: Quotation enquiry *Message*: send Enquiry message *Other party task*: Begin *type*: simple performance *Order*: none *node*: Evaluate Quotation *Message*: receive Quotation *Other party task*: Replay Quotation *type*: simple *Order*: none	*node*: Purchase Order *Message*: 1. send Order Form *Other party task*: Replay Quotation *type*: obligatory performance Message: 2. receive Domain Name *Other party task*: Domain Name Registration *type*: obligatory *Order: 1<2.*

From the above example, we can see that since there is no centralized control, each party of the contract defines the communicating tasks so that they can receive and send messages appropriately, thus implementing the specified communication. Because the *e-contract communication graph* is very important, an additional check can be executed to capture any accidental inconsistencies. The consistency-checking algorithm verifies that the communication links are defined as specified. For example when a part sends a message the other party has confirmed that it is waiting to receive the message and they both agree on the type of the constraint.

4.3 Enforcing Contracts

With workflow views support, the enforcement of contracts among different organizations can be facilitated. Examples include the following:

Installation schedule – Dickson Computer System can compute an installation schedule to the end-user according to reported lead-time of the computer vendor and the lease-line provider. If a vendor changes the lead-time, but the installation schedule can still be completed within the end-user's deadline, the change can be tolerated. Otherwise, another source has to be sought for, or an alternative solution should be employed, subject to the end-user's approval.

Price - If the price for the web-server rises to an extent that there are no more profits, Dickson Computer System may want to request an increase in price, use of an alternate cheaper server, delay the delivery until the price drops, or cancel the order. However, as protected by a contract, the end-user has the right to enforce the contract.

Server availability - If a certain server, or critical part of the server is stopped from production, Dickson Computer System may request the end-user's approval of using an alternative server or part.

Because all the important information critical to the enforcement of the contract, viz. availability, price, lead-time, etc., are available in various workflow views and noticeable through effective E-ADOME event-triggering mechanisms, contracts can be maintained in an effective inter-organizational workflow environment.

4.4 Discussion

The section has presented an overview of managing view-based E-contracts in a cross-organization E-service workflow environment. The management of E-contracts is greatly facilitated by the workflow view mechanism for security, information hiding, workflow adaptation, providing different interactions with different organizations, and e-contract enforcement.

5 Related Work

While the concept of workflow view is novel, our approach has been motivated by views in object-oriented data models which can be dated back to [8], and in particular by imaginary objects in [1]. [10] discusses federated OODBMS and views for objects in a distributed environment.

Dartflow [2] is one of the first web-based WFMS, using transportable agents, CGI and Java technologies. Eflow [3] is one of the closest commercial systems with features like E-ADOME in handling e-Services. However, Eflow does not address matching of agents directly with tasks. Instead, it uses the concept of generic service node and service selection rules. Currently, several commercial WFMSs such as TIB/InConcert [18] and Staffware 2000 [16] provide web user interface too. In addition, I-Flow [9] has a Java workflow engine. WW-flow [9] provides a hierarchical control scheme over workflows implemented in Java for both the workflow engine and client interfaces. It allows sub-workflows to be executed in different workflow engines across the web. As for standards, Workflow Management Coalition (WfMC) has recently proposed Wf-XML [22], which is an interchange format specification for an XML language designed to model the data transfer requirements for process specification.

It is a new approach to E-service enactment based on an advanced WFMS engine. Besides E-ADOME, other notable systems using related approaches include Eflow [3] and Crossflow [11]. Crossflow models virtual enterprises based on a service provider-consumer paradigm, in which organizations (service consumers) can delegate tasks in their workflows to other organizations (service providers). Virtual organizations are dynamically formed by contract-based matchmaking between service providers and consumers. Though Crossflow includes detailed work for contracts, contract enforcement is also not so straightforward as the support provided by E-ADOME workflow views equipped ECA-rules mechanisms based on cross-organizational events. [19] presents workflow schema exchange in an XML dialect called “XRL”.

However, few of the above-mentioned WFMSs support web-based cooperative exception handling. Most of them contact clients based on electronic mail and web forms and does not directly support active paging of clients with Internet message facilities like ICQ [20]. Very few commercial WFMSs provide support for handling exception. Even if they do, they only address very basic problems in a slight extend. It is also a new approach to build E-service agents based on an advanced WFMS engine. Besides E-ADOME, other notable systems using related approaches include Eflow [3] and Crossflow [11]. However, E-ADOME has the richest features in coordinating distributed agents than other systems close to us. Further details in the novelty of ADOME-WFMS, especially in its pragmatic meta-modeling approach and exception-handling features, are presented in [6,7].

6 Conclusions

This paper has presented an advanced cross-organizational workflow environment with novel features in cooperating with other organizations over the Internet for E-service enactment. We have illustrated in the context of E-ADOME, how its ADOME-WFMS engine, a flexible WFMS based on ADOME active OODBMS with role and rule facilities, is extended to accomplish such objectives. Compared with other research on this topic, E-ADOME provides an improved environment for various types of process enactment, which can adapt to changing requirements, with extensive support for reuse. This paper has introduced a novel concept of workflow view for interfacing different WFMSs, possibly belonging to different organizations, and its applications in an e-service environment. We have proposed a contract model based on workflow views, to simplify the process of developing cross-organizational workflow regarding to contracts. We have also illustrated how management of E-contracts is greatly facilitated by the workflow view mechanism for security, information hiding, workflow adaptation, providing different interactions with different organizations, and e-contract enforcement. Further note that, E-ADOME specification of workflows is based on standardized Workflow Management Coalition workflows, many of the techniques presented in this article can be applicable to any WFMSs for E-service enactment.

We are working on further details of process adaptation for interoperability, e-contract negotiation, methodologies for e-contract enforcement (including preventive measures), based on cross-organization workflows and the workflow view mechanism. We consider further research issues on interfacing and interoperability important for extending the applicability of an advanced WFMS engine. We are interested in the application of E-ADOME in various advanced real-life e-commerce environments, such as procurement, finance, stock trading and insurance. We are developing a more unified way to exchange information, including workflow views, with other agents, with XML. ADOME is currently being built on top of the ADOME-WFMS prototype system, with a web-based user interface to accommodate the whole range of activities.

References

1. Abiteboul, S. , Bonner, A. : Objects and Views. In Proceedings of ACM SIGMOD Conference, 1991
2. Cai, T., Gloor, P., Nog, S.: DartFlow: A Workflow Management System on the Web using Transportable Agents, Technical Report PCS-TR96-283, Dartmouth College, Hanover, N.H., (1996)
3. Casati, F., et al.: Adaptive and Dynamic Service Composition in eFlow. HP Laboratories Technical Report HPL-2000-39 (2000)
4. Chiu, D.K.W., Karlapalem, K., Li, Q.: E-ADOME: A Framework For Enacting E-services. VLDB Workshop on Technologies for E-Services, Cairo, Eygpt (2000)
5. Chiu, D.K.W., Karlapalem K., Li Q.: Views for Inter-Organization Workflow in an E-Commerce Environment, 9th IFIP 2.6 Working Conference on Database Semantics (DS-9), Hong Kong (2001)

6. Chiu, D.K.W., Li, Q., Karlapalem, K.: A Meta Modeling Approach for Workflow Management Systems Supporting Exception Handling, Special Issue on Method Engineering and Metamodeling, Information Systems, Elsevier Science, 24(2) (1999)159-184
7. Chiu, D.K.W., Li, Q., Karlapalem, K.: Web Interface-driven Cooperative Exception Handling in ADOME Workflow Management System, Information Systems, Elsevier Science, 26(2) (2001) 93-120
8. Dayal, U.: Queries and Views in an Object-Oriented Data Model. In Proceedings of 2nd International Workshop on Database Programming Languages (1989)
9. Enix Consulting Limited: An Independent Evaluation of i-Flow Version 3.5 (2000) (available at http://www.i-flow.com).
10. Gardarin, G., Finance, B., Fankhauser, P.: Federating object-oriented and relational databases: the IRO-DB experience. In Proceedings of the 2nd IFCIS International Conference on Cooperative Information Systems (CoopIS '97), (1997) 2-13
11. Grefen, P., Aberer, K., Hoffner, Y., Ludwig, H.: CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises. International Journal of Computer Systems Science & Engineering, 15(5) (2000) 277-290
12. Kim, Y., Kang, S., Kim, D., Bae, J., K. Ju: WW-Flow: Web-Based Workflow Management with Runtime Encapsulation. IEEE Internet Computing, 4(3) (2000) 56-64
13. Li, Q., Lochovsky, F. H.: ADOME: an Advanced Object Modeling Environment. IEEE Transactions on Knowledge and Data Engineering, 10(2) (1998) 255-276
14. McCarthy, D., Sarin, S.: Workflow and Transactions in InConcert. IEEE Data Engineering,16(2) (1993) 53-56
15. Miller, J., Sheth, A., Kochut, K., Luo, Z.: Recovery Issues in Web-Based Workflow. Proceedings of the 12th International Conference on Computer Applications in Industry and Engineering (CAINE-99), pp. 101-105, Atlanta, Georgia (1999)
16. Staffware Corporation: Staffware Global - Staffware's Opportunity to Dominate Intranet based Workflow Automation (2000) http://www.staffware.com
17. ter Hofstede, A., Orlowska, M., Rajapakse, J.: Verification Problems in Conceptual Workflow Specifications. Data & Knowledge Engineering, Pergamon Press, Elservier Science, 24(3) (1998) 239-256
18. TIBCO Software Inc., which has acquired InConcert Inc., http://www.tibco.com
19. van der Aalst, W.M.P., Kumar, A.: XML Based Schema Definition for Support of Interorganizational Workflow. In Proc. 21st International Conference on Application and Theory of Petri Nets (ICATPN 2000), Aarhus, Denmark (2000)
20. Weverka,. P.: Mastering Icq: The Official Guide. IDG Books. ICQ Press (2001).
21. Workflow Management Coalition: The Workflow Reference Model. (WFMC-TC-1003, 19-Jan-95, 1.1) (1995)
22. Workflow Management Coalition: Workflow Standard – Interoperability Wf-XML Binding, WFMC-TC-1023 (2000)