Workflow View Driven Cross-Organizational Interoperability in a Web-Service Environment

Dickson K.W. Chiu[1], Shing-Chi Cheung[2], Kamalakar Karlapalem[3], Qing Li[4], and Sven Till[2]

[1] Department of Computer Science and Engineering, Chinese University of Hong Kong
kwchiu@acm.org
[2] Department of Computer Science, Hong Kong University of Science and Technology
{scc,till}@cs.ust.hk
[3] International Institute of Information Technology, Gachibowli, Hyderabad 500 019, India
kamal@iiit.net
[4] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
csqli@cityu.edu.hk

Abstract. In an E-service environment, workflow involves not only a single organization but also a number of business partners. Therefore, workflow interoperability in such an environment is an important issue for enacting workflows. In this paper, we introduce our approach of using workflow views as a fundamental support for E-service workflow interoperability and for controlled (sub-) workflows visibility to external parties. We present a meta-model of workflow views and their semantics with example usage. We develop an interoperation model based on workflow views, with a supply-chain E-service cross-organization workflow example. We also propose an implementation of workflow view and cross-organizational interoperability based on contemporary Web service [14] technology, with respect to our E-ADOME workflow engine.

Keywords: e-service, cross-organizational workflow, workflow management, workflow views, Web service, interoperation protocol

1 Introduction

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 workflow management systems (WFMSs) are now web-enabled (such as [5], [7], [11], [19], [28], [33], [34], [38]) and recent researchers in workflow technologies are exploring cross-organizational workflows to model these activities. We have proposed a novel

Ch. Bussler et al. (Eds.): WES 2002, LNCS 2512, pp. 41–56, 2002.

approach of applying *workflow view* for supply-chain management and e-service enactment [11] in a cross-organizational workflow environment. As follow-up work, we detail in this paper how workflow views can be implemented with contemporary Web service [14] technology, with respect to our E-ADOME workflow engine extended with agent interfaces.

Views help balance trust and security, that is, only information necessary for the process enactment, enforcement and monitoring of the service is made available to both parties, in a fully controlled 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 supported by E-ADOME [11]) is only required upon their first interaction, and reusable subsequently, unless their workflows are changed drastically. Because an organization is probably interoperating with many other different organizations, different views of a workflow can be presented to different organizations according to different requirements. Since the E-service arena is very competitive, cross-organization workflows can be developed fast and managed adequately, together with e-protocols.

The contribution and coverage of this paper are as follows: (i) a cross-organization workflow approach for a composite E-service with the concept of workflow views, (ii) a cross-organizational interoperation model based on workflow views, (iii) details on the facilitation of workflow views and cross-organizational interoperability with contemporary Web service technology, (iv) demonstration of the applicability of E-ADOME in supporting E-service through these features.

The rest of our paper is organized as follows. Section 2 presents a motivating example to illustrate the concept of workflow views in an E-service cross-organizational workflow environment. Section 3 presents our view-based model for e-protocols. Section 4 illustrates how workflow views facilitate e-protocol management, such as e-protocol definition and enforcement, in an E-service environment. Section 5 presents the E-ADOME architecture to illustrate how a flexible WFMS engine can be extended to coordinate distributed agents. Section 6 compares related work. Finally, we conclude this paper with our plans for further research in Section 7.

2 Motivating Example

In this section, we present a motivating example of cross-organization workflow based on a supply chain e-commerce scenario, as depicted in Fig. 1. There are three types of organizations involved, viz., end-users, system integrators, and parts vendors. Each of the individual workflow is simple, but the cross-organizational interactions are more interesting and complicated.

The end-user undergoes a requisition workflow, say, for an advanced server system. First, quotation enquiries are sent to a number of system integrators. The received quotations with product information are evaluated. A purchase order is then issued to the selected system integrator. The server system is then received and checked. Finally, payment is arranged for.

A system integrator's workflow starts when an enquiry is received. The first step is to check from its parts vendors the lead-time and updated price of major parts, especially for those with a large price fluctuation (e.g., CPU and memory). After evaluation, a quotation is sent to the end-user. While the end-user evaluates the quotation, the system integrators may need to provide additional or updated information for the bid. After a purchase order is received, the successful system integrator then orders necessary missing parts that are not in stock, and estimates a delivery schedule. When all the parts are ready, the system integrator assembles, tests the server, and then delivers it. Finally, after receiving the payment, the workflow ends.

A parts vendor's workflow also starts when an enquiry is received. Assuming this is the end of the supply chain, the vendor has all necessary information to reply the system integrator with updated parts information and prices. Assuming that B-to-B orders on standard parts are usually performed together with payment, this workflow ends after the delivery of the ordered parts.

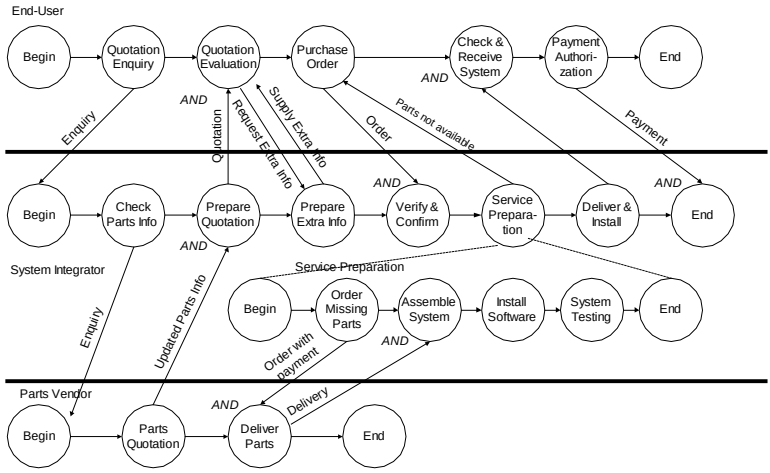

Fig. 1. Cross-Organizational Workflow of a Supply-Chaining Example

3 A Meta-model for 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 interoperating and interacting workflows from different organizations). This is known as cross-organizational workflow. To support workflow interoperability, 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 is a structurally cor-

rect subset of a workflow definition (as in [19]). We propose to use the concept of workflow views (which is detailed in the next section) to help advanced interactions among WFMSs and allow them to interoperate in a gray box mode (i.e., they can access each other's internal information to some extent). Therefore, workflow views can provide a handy mechanism to support E-protocol enactment and enforcement across organizational boundary over the Internet.

On the other hand, 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 in which 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).

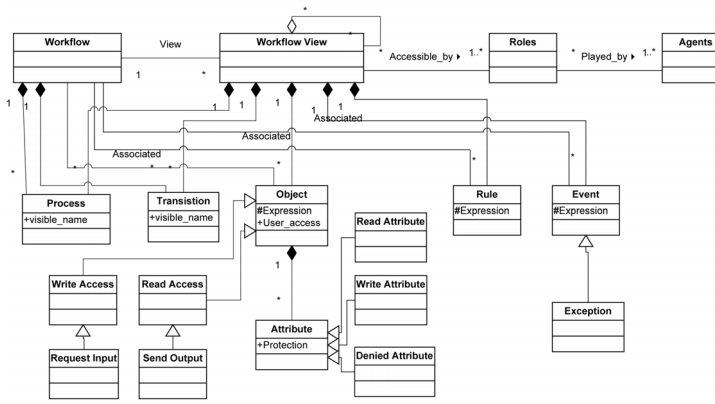

Fig. 2. Workflow View Model in UML

```
view v of workflow w begin
   {process p1 view v1 ...}
   {process p2 renames p3 ...}
   {transition 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, ...}
   {access role1, role2, ... }
end
```

Fig. 3. Workflow View Definition Language

The components of a workflow view include the process flow graph, input/output parameters, objects, rules, events, exceptions and exception handlers derived from the original workflow. A detail description of these components is available in [11]. Fig. 2 depicts our workflow view model new refined with Unified Modeling Language (UML) while Fig. 3 depicts a simple workflow view definition language in accordance with our model. The XML schema of the language is given in [9].

We have also recently added *Access Security Control* to workflow views. Each workflow view must be specified with one or more accessible *roles*. A role represents a collection of agents of similar properties [32] and therefore can also be used in specifying security context. The role concept is reminiscent of the "group ids" and "group access rights" of a UNIX system and also of the security model used in Enterprise JavaBeans [37]. Different partners may belong to different roles so that they possess different access rights. This leads them to different views of the same business process. The role concept serves the purpose of classifying business partners into groups, enabling personalized business processes. A party belonging to a certain role or to several roles exercises its access rights at the view level according to the rights assigned to the specified roles.

Fig. 4 depicts two example workflow views of the end-user and Dickson Computer Systems respectively. These two workflow views are derived from their workflows depicted in Fig. 1. An XML version of the example is given in [9]. They will be further used in subsequent sections for illustrating the business process interoperation between these two parties. Fig. 6 also presents these two views graphically.

```
view E1 of workflow End_user begin
  process Quotation_enquiry, Evaluate_quotation, Purchase_order,
          Check_rec_system view C1, Payment_authorization view P1;
  object enquiry output, quotation input, extra_info request, order_form output,
         installation_and_delivery input, ...;
  event Change_schedule, ...;
  exception Payment_delay, ...;
  access supplier;
end

view D1 of workflow Dickson_computer_systems begin
  process Check_parts_info, Prepare_quotation, Prepare_Extra_Info,
          Verify_and_confirm view V1, Service_preparation view S1,
          Order_missing_parts view O1, Assemble_system view A1,
          Install_software view I1, System_testing view S2,
          Deliver_and_Install view D1;
  object enquiry input, quotation output, extra_info, order_form input,
         installation_and_delivery output, ...;
  event Change_schedule, Change_price, ...;
  exception Parts_not_available, ...;
  access customer;
end
```

Fig. 4. Workflow Views Example

4 Cross-Organization Interoperability with Workflow Views

In this section, we present a model for cross-organizational workflow interoperability based on workflow views, and show how this model can be facilitated with Web services [14].

4.1 Cross-Organizational Interoperation Model Based on Workflow Views

Based on the workflow view mechanism described in the previous section, we now proceed to describe its application in the domain of e-services, particularly, the cross-organization interoperation model. As depicted in Fig. 5, an *interoperation protocol*

consists of workflow views, communication graphs between these views, and a set of interoperation parameters. This information should be stored at each of parties involved in the business process. In this paper, we concentrate on the situations of involving two parties only, because these are the (basis for) majority of interoperation scenarios. Furthermore, business processes involving multiple parties are often having pair-wise interactions only (e.g., end-user with the supplier, supplier with the part vendor, but not end-user with the part vendor directly).

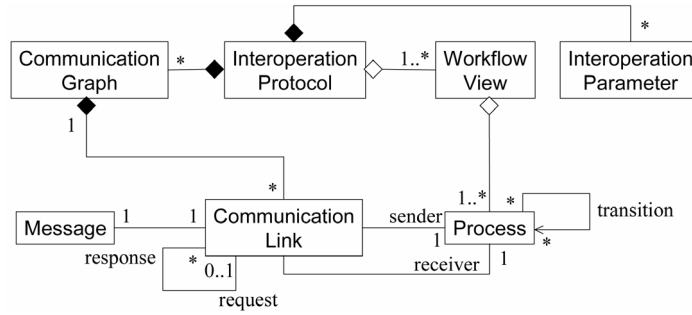

Fig. 5. UML Model of Interoperation Protocol Based On Workflow Views

Every business process has some basic information to be captured by an information system. In our interoperation model, the interoperation parameters capture a set of attributes whose values describe the necessary information for the business process, which is usually in the form of parameters. Example attributes can be *Accept*, *Offer*, *Goal*, *Schedule*, *Payment*, *Documents*, *QoS*, *Exception_Rules*, *Commit*, etc. During interoperation of the business process, besides the parameters, the two parties have to agree on a common workflow, task assignments, and cross-organizational message exchanges. For example, in Fig. 6 we have a protocol between Dickson Computer Systems and the end-user. Each party has its own internal workflow. In order to interoperate, each party must be able to view a subset of the other party's workflow that will specify the tasks obliged to perform. A key issue here is that in every protocol we have to balance two concepts: trust and security. When two parties interoperate, we assume that there is trust between them and that information necessary for the specification, enforcement and monitoring is available to both parties. At the same time, for security reasons no party wants to reveal information more than necessary to the other party. In our workflow protocol model, the balance is achieved through the 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 level 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 cross-organizational

communications in order to support the specific business process. These communications are useful for information exchange, control exchange, and synchronization. In our interoperation model, there are some tasks in each workflow view, called *communicating tasks*, through which two parties communicate. The cross-organizational control and information flow is specified within *communicating tasks* and their associated *communication links*. We associate the message with a label of the communication link. Furthermore, each *communicating task* receives and sends a set of messages, the order in which these messages occur is crucial. Therefore, with every *communicating task*, we impose on the messages a partial order that has to send/receive. For example in Fig. 6, the *Quotation Evaluation* task of the end-user's workflow has to interact with the *Prepare Quotations* and *Prepare Extra Info* tasks of Dickson Computer Systems. As a result, the end-user receives a *QuotationResponse*. In addition, synchronization achieved by cross-organizational messages is label with "AND" in the graph, to represent an *and-join*. For example, the *Quotation Evaluation* task can only be started upon receiving the *Quotation* message from Dickson Computer Systems.

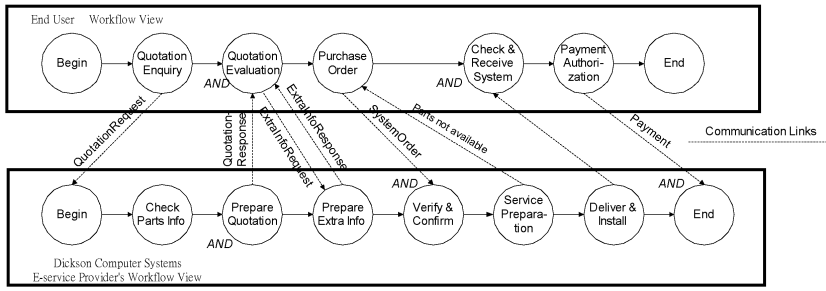

Fig. 6. A Communication Graph of an Interoperation Protocol between Two Workflow Views

4.2 Specification of the Interoperation Model with XML

After two parties have decided to make an interoperation, they have to arrive at an interoperation protocol, which specifies the details. When two parties want to form a protocol, first they have to decide on the value of the *interoperation parameters*, like the following example based on Fig. 6.

The description (cf. Fig. 7) is the proof that both parties have agreed on the formation of a protocol, and the interoperation protocol model depicts the details. An XML representation of the example is given in [9]. Then, each party has to present the view as specified in the interoperation protocol model, in order to allow access to workflows of each other, and to incorporate the protocol requirements on the data and control flow. (Fig. 4 depicts an example of the views employed in this protocol.) As these workflow views specify the data output and input requirements of both parties in the object statement, both parties can identify the communicating tasks of both workflow views. Then, by matching these input and output messages, both parties

can derive the necessary communication links and their order. At the same time, this process also detects possible mismatches. Fig. 8 lists some example communication tasks and their links. The XML representation of the example is given in [9].

```
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
   ...
```

Fig. 7. An Interoperation Interface Example

node: Quotation Enquiry
- *Message*:send QuotationRequest
  Other party task: Begin
	node: Quotation Evaluation
	- *Message*: receive QuotationResponse
	  Other party task: Verify and Confirm
	- *Message:* send ExtraInfoRequest
	  Other party task: Prepare Extra Info
	- *Message:* receive ExtraInfoResponse
	  Other party task: Prepare Extra Info

Fig. 8. Some example communication tasks and their links

From the above example, we can see that since there is no need for centralized control, each party of the protocol defines the communicating tasks so that they can receive and send messages appropriately, thus manifesting the required communications.

4.3 Design of Web Service Interfaces for Interoperability

Web services [14] provide a new interoperable platform for Internet applications. These applications typically offer self-contained and self-describing services that can be published, located, and invoked across the Internet. Web services perform functions, which can be anything from simple requests to complicated processing of business data. Once a Web service is deployed, other applications and Web services can discover and invoke the Web services based on the technologies that support: (i) a mechanism to register a service; (ii) a mechanism to find a service; and (iii) a mechanism for two parties to communicate. A service can be invoked by either an application call or a HTTP request through a Web-based interface. This ensures that software

systems can be coupled at different application levels. The Web service architecture is suitable for cross-organizational collaboration in a highly dynamic environment as it supports just-in-time integration, encapsulating and true interoperability. This allows the implementations to be programming language-neutral and communications mechanism-independent.

Name: QuotationService:
Location/Provider: System Integrator
Input: QuotationRequest
- CustomerInformation
  - Name
  - Address
  - Customer number
- SystemConfiguration
  - NumberOfUnits
  - Part List
  - Dimensions

Output: QuotationResponse
- Quotation
  - CustomerInformation
  - SystemConfiguration
  - ItemizedPriceList
  - TotalDiscountedPrice
  - ShippingInformation
  - TermsOfServices
  - ExpiryDateOfOffer
  - DeliveryDate

Name: reqPartsQuotation:
Location/Provider: Parts Vendor
Input: PartsQuotationRequest
- CustomerInformation
  - Name
  - Address
  - Customer number
- RequestedPart
  - Part number
  - Price
  - Dimensions
  - NumberOfUnits

Output: PartsQuotationResponse
- Quotation (Pricelist)
  - Service
  - Price
  - Components

Name: reqExtraInfo
Location/Provider: System Integrator
Input: ExtraInfoRequest
- Customer number
- Customer Name
- Extra info request number
- Quotation/offer number
- Quotation date
- Request date
- Questions
  - Question number
  - Reference to quotation
  - Intrinsic question
- Requested response time

Output: ExtraInfoResponse
- Supplied extra info
  - Extra info request number
  - Extra info request date
  - Answers:
    - Questions number
    - Intrinsic answer

Name: payInvoice
Location/Provider: SystemIntgrator
Input: Payment
- Invoice number
- Invoice date
- Payer
- Payee
- Invoice amount

Output:PaymentAcknowledgement
- result (Boolean) successful/ unsuccessful

Name: orderSystem
Location/Provider: System Integrator
Input: SystemOrder
- Customer number
- Parts quotation/offer number
- Delivery address
- Requested delivery date
- Amount
- Total price

Output: OrderConfirmation
- Order successful/ unsuccessful
- Invoice
  - Invoice number
  - Invoice date
  - Due date of payment
  - Bank account info
- Scheduled delivery date

Name: orderParts
Location/Provider: Parts vendor
Input: PartsOrder
- Customer number
- Parts quotation/offer number
- Delivery address
- Requested delivery date
- Article number
- Number of articles
- Amount
- Article price
- Total price
- Credit card number/ or other proof of payment

Output: PartsDelivery
- Order successful/ unsuccessful
- Bill
- Scheduled delivery date

Fig. 9. Some Web Services to be Provided by Dickson Computer Systems

Web services are described through their interface definitions in the Web Service Description Language (WSDL [43]), which is an XML-based language developed to address the following questions about a Web service: (i) What does the service do? (ii) How can it be accessed? and (iii) Where can it be accessed? These three questions are mapped to the abstract specification of a service, a specific implementation of that service, and the location of the service implementation. The implementation of Web Service interfaces is based on the exchanged messages. Especially the content and the structure of the interfaces reflex the exchanged business entities. Fig. 9 presents a set

of possible Web services implemented by Dickson Computer Systems to support the incoming external events and their immediate responses that are identified from the data requirement analysis of the supply-chain example. Appendix E gives the definition of the QuotationService Web Service in *Web Service Description Language (WSDL)*. In this example, the incoming events and their immediate responses are handled by one Web service. For example, QuotationRequest is the input message and the QuotationResponse the output message of the Web service QuotationService. The similarity between the input and the output messages of a communication task and the interface definition in the Web service description provides a hint to the derivation of potential Web services, especially due to the fact what all the message exchanges are already represented in XML format. [8] describes not only how the Web services can be derived in details, but also how they can be used to provide workflow extensions for e-services across organizations.

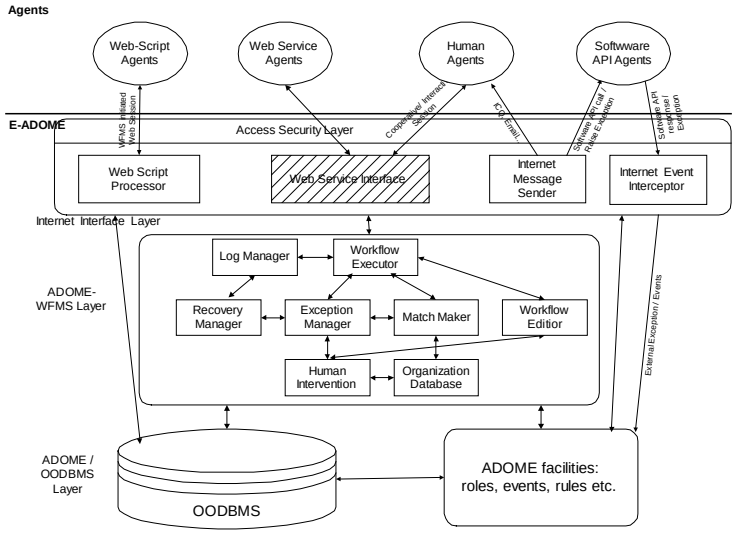

Fig. 10. E-ADOME Architecture

5 E-ADOME Architecture Enhanced with Web Service

We extend a flexible, web-enabled workflow management system, ADOME-WFMS [12][13], into E-ADOME to provide support for specifying, executing and monitoring composite e-services. In particular, we strengthen the external interface layer to interact with different types of agents over the Internet more effectively. The most recent update is the employment of Web service support [14] to replace a traditional web-server. Because agents over the Internet probably originate from different organizations and often operate with different interfaces, we cannot change foreign public agents (such as their web pages) to the exact interface we want. Instead, adaptations are needed to accommodate for them. As shown in Fig. 10, the E-ADOME environment can be divided into the following layers:

ADOME / OODBMS Layer – ADOME was developed to enhance the knowledge-level modeling capabilities of OODBMS models [28], so as to allow them to more adequately deal with data and knowledge management requirements of advanced information management applications, especially WFMSs. *ADOME-WFMS Layer* – this is a flexible WFMS built upon ADOME facilities, supporting effective management of agents, on-line workflow evolution, and automatic and cooperative exception handling [12].

Internet Interface Layer – this is the enhancement layer to the WFMS for ADOME-WFMS to interact with various types of external agents through the Internet. *Internet Message Sender* sends alerts to users and agents via ICQ or E-mail; this module also sends out requests to other software agents using a compatible API. *Internet Event Interceptor* receives responses or alerts from software agents through a compatible API and translates them to ADOME events (which include exceptions). Note that an agent may be internal or external to the organization and may itself be another WFMS. Furthermore, an *Access Security Layer* is added to handle external communications. *Web Script Processor* enables the E-ADOME to initiate an automatic conversation script with other interactive, web-based service providers without compatible software API, including most on-line ordering web pages or service report forms. (Without this facility, the WFMS would need a staff to perform this task manually.)

The newly added *Web Service Interface* module enables E-ADOME to communicate with other WFMSs to allow for more advanced task execution and control in foreign WFMSs. It enables human agents or users to interact with E-ADOME, to access the database, and to report work progress, in addition to programmed web interface. The *Web Service Interface* module can be integrated without changing of underlying layers.

The enabling power for e-service applicability mainly relies on the additional Internet interface layer on top of ADOME-WFMS. This interface layer can send and receive messages through the Internet, in order to communicate with distributed users and other service agents. Arrival of incoming messages can be detected as events to trigger actions of the WFMS specified by both regular workflow specification and Event-Condition-Action rules. In particular, the WFMS interface module, now equipped with Web Service mechanism, further facilitates cross-organizational e-service (workflow) enactment. As the E-ADOME extension is built on top of ADOME-WFMS, it is perceived that the techniques presented in this paper are applicable to other similar WFMSs or other information systems.

6 Related Work

Modeling of interoperation protocols can be dated back to the Protocol Net Protocol [35]. However, they only concentrated on low-level transaction aspects. [19] presented a framework for legal protocols, but not a mechanism for modeling interoperation protocols. [31] described a model for representing electronic trade procedures based on Petri Nets. [22] introduced a declarative approach to business rules in e-

commerce protocols by combining Courteous Logic Program and XML. Recently, [27] proposed a meta-model for interoperation protocols with E-R diagrams and generation of workflows to support interoperation protocols, but did not consider the notion of workflow views and the notion of commitment in interoperation protocols. The COSMOS project [24] developed an Internet-based electronic protocol service based on XML and CORBA components, but not based on workflow technologies.

Our preliminary approach of workflow views has been presented in [11]. This approach has been motivated by views in object-oriented data models, which can be dated back to [15], and in particular by imaginary objects in [1]. [18] discusses federated OODBMS and views for objects in a distributed environment. [29] presented an algorithm for workflow view construction and verification, but did not discuss any of its applications. [2] introduced the concept of inheritance of a public workflow from a private workflow to achieve interoperability in a cross-organizational e-commerce environment.

Dartflow [5] is one of the first web-based WFMS, using transportable agents, CGI and Java technologies. WebWork [33] described some issues in web-based workflow recovery, but only on WFMS and web related failures without covering user-defined workflow exceptions. Eflow [7] 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 [38] and Staffware 2000 [34], provide web user interfaces too. In addition, I-Flow [17] has a Java workflow engine. WW-flow [28] 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.

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 [7] and Crossflow [19]. 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). Though Crossflow includes detailed work for protocols, it does not provide such a sophisticated mechanism as workflow views for information and control exchange between workflows of different organizations. [1] presents workflow schema exchange in an XML dialect called “XRL” but does not include support for workflow views.

As for standards, Workflow Management Coalition (WfMC) has recently proposed Wf-XML [39], which is an interchange format specification for an XML language designed to model the data transfer requirements for process specification. Meanwhile WfMC is working towards an industrial standard with the WfMC Reference Model [40] for WFMS so as to identify their characteristics, terminology [41] and components, and to enable the individual specification to be developed within the context of an overall model for WFMS. However, only very recently, WfMC published a white paper on event extension to WFMS [42], but they still do not specifically address exceptions.

An upcoming standard for B2B integration is electronic business XML (ebXML) [16]. The proposed framework contains the idea that two trading partners agree on a collaboration protocol, which contains the messaging service interface requirements as well as the implementation details pertaining to the mutually agreed upon business processes. However, the paradigm of workflow views is more general. It provides mechanisms to bridge the external interfaces and the internal workflows of a business party in a controlled way. ebXML can be used, among other languages, to implement workflow view details for establishing cross-organizational cooperation.

In summary, not all the above-mentioned WFMSs support various kinds of interactions with different kinds of agents, as in the E-ADOME interface layer. Very few commercial WFMSs provide support for handling exception and workflow adaptation comprehensively. Compared with the systems close to ours, E-ADOME has the most features available to support E-services, interoperation protocols, and mobile agents on the Internet.

7 Conclusions

This paper has presented an advanced cross-organizational workflow environment with pragmatic 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. We have also detailed the employment of contemporary Web service technology in specifying and enacting E-services with the workflow view support.

Compared with other research efforts 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 the use of workflow view for interfacing different WFMSs, possibly belonging to different organizations, and its applications in an e-service environment. We have proposed an interoperation model based on workflow views, to simplify the process of developing cross-organizational workflows. We have also illustrated how cross-organizational business processes can be greatly facilitated by the workflow view mechanism for security, information hiding, workflow adaptation, providing different interactions with different organizations. Moreover note that, E-ADOME specification of workflows is based on standardized Workflow Management Coalition workflows, many of the techniques presented in this paper can thus be applicable to any WFMSs for E-service enactment.

For workflow views, we are working on further details of formal definitions, construction and verification algorithms, more detailed taxonomy, view update mechanisms, and more operations support. For interoperation protocols, we are working on further details of process adaptation for interoperability, multiple-party protocols and sub-protocols, interoperation protocol negotiation, a more comprehensive methodology for interoperation protocol enforcement. We are also interested in the application of E-ADOME into various advanced real-life e-commerce environments, such as

procurement, finance, stock trading and insurance. We are also interested in wrappers to interface with legacy software agents. E-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. W.M.P. van der Aalst and A. Kumar. XML Based Schema Definition for Support of Inter-organizational Workflow. In Proc. 21st International Conference on Application and Theory of Petri Nets (ICATPN 2000), Aarhus, Denmark (2000)
2. W. M. P. van der Aalst, M. Weske. The P2P Approach to Interorganizational Workflows. In Proceedings of 13th International Conference Advanced Information Systems Engineering (CAiSE 2001), Interlaken, Switzerland, June 2001, Springer LNCS 2068, pp140-156.
3. S. Abiteboul and A. Bonner. Objects and Views. In Proceedings of ACM SIGMOD Conference, 1991.
4. G. Alonso, et al. Exotica/FMDC: a workflow management system for mobile and disconnected clients. *Distributed & Parallel Databases*, **4**(3):229-247, 1996.
5. Ting Cai, Peter A. Gloor, Saurab Nog, DartFlow: A Workflow Management System on the Web using Transportable Agents, Technical Report PCS-TR96-283, Dartmouth College, Hanover, N.H., 1996.
6. F. Casati, G. Pozzi. Modeling Exceptional Behaviours in Commercial Workflow Management Systems. In *Proceedings of the 4th International Conference on Cooperative Information Systems (IECIS 98)*, IEEE Press, 1998.
7. F. Casati, et al. Adaptive and Dynamic Service Composition in eFlow. HP Laboratories Technical Report HPL-2000-39, March 2000.
8. S.-C. Cheung, D.K.W. Chiu and S. Till. *A Data-Driven Approach to Extending Workflows Across Organizations over the Internet* Technical Report HKUST-CS04-02, Hong Kong, February 2002
9. D.K.W. Chiu, S.C. Cheung, K. Karlapalem, Q. Li AND S. Till: *Workflow View Driven Cross-Organizational Interoperability in a Web-Service Environment* Technical Report HKUST-CS17-02, Hong Kong, May 2002
10. D.K.W. Chiu, K. Karlapalem and Q. Li. E-ADOME: A Framework for Enacting E-services. *VLDB Workshop on Technologies for E-Services*, Cairo, Eygpt, Sept. 2000.
11. D.K.W. Chiu, K. Karlapalem and Q. Li. Views for Inter-Organization Workflow in an E-Commerce Environment, *9th IFIP 2.6 Working Conference on Database Semantics (DS-9)*, Hong Kong, April 2001.
12. D.K.W. Chiu, Q. Li and K. Karlapalem,. A Meta Modeling Approach for Workflow Management System Supporting Exception Handling. *Information Systems*, Pergamon Press, Elservier Science, 24(2):159-184, 1999.
13. D.K.W. Chiu, Q. Li and K. Karlapalem. Web Interface-Driven Cooperative Exception Handling in ADOME Workflow Management System. *Information Systems*, Pergamon Press, Elservier Science, 2001.
14. V. Chopra, et. al. Professional XML Web Services, Wrox Press, 2001.
15. U. Dayal. Queries and Views in an Object-Oriented Data Model. In Proceedings of 2nd International Workshop on Database Programming Languages, 1989.
16. http://www.ebXML.org

17. Enix Consulting Limited. An Independent Evaluation of i-Flow Version 3.5, 2000 (available at http://www.i-flow.com).
18. G. Gardarin, B. Finance and P. Fankhauser. 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.
19. M. Gisler, K. Stanoevska-Slabeva, and M. Greunz, Legal Aspects of Electronic Protocols, *In CAiSE*00 Workshop of Infrastructures for Dynamic Business-to-Business Service Outsourcing (IDSO'00)* Stockholm, 5 - 6 June 2000.
20. P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig; CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises; International Journal of Computer Systems Science & Engineering, 15(5):277-290, 2000.
21. F. Griffel, et al. Electronic Protocoling with COSMOS – How to Establish, Negotiate and Execute Electronic Protocols on the Internet. *2nd Int. Enterprise Distributed Object Computing Workshop (EDOC '98)*, 1998.
22. B. N. Grosof, A declarative approach to business rules in Protocols: Courteous Logic Programs in XML, *Proceedings of the 1st ACM Conference on Electronic Commerce (EC99)*, Denver, Colorado, USA, Nov. 3-5, 1999.
23. ter Hofstede, M. Orlowska and J. Rajapakse. Verification Problems in Conceptual Workflow Specifications. Data & Knowledge Engineering, Pergamon Press, Elservier Science, 24(3) (1998) 239-256
24. Hewlett Packard. Changengine Admin Edition (AdminFlow) Process Design Guide, 1998.
25. Itasca Reference Manual, Ibex Corporation, 1994.
26. ICQ. http://www.icq.com
27. K. Karlapalem, A Dani and P. Krishna. A Frame Work for Modeling Electronic Protocols. Proceedings of the 20th International Conference on Conceptual Modeling, Yokohama, Japan, Nov 2001, Springer LNCS 2224, pp193-207.
28. Y. Kim, S. Kang, D. Kim, J. Bae, and K. Ju. WW-Flow: Web-Based Workflow Management with Runtime Encapsulation. IEEE Internet Computing, 4(3):56-64, 2000.
29. D.-R. Liu and M. Shen. Modeling Workflows with a Process-View Approach, Proceedings of the 7th International Conference on Database Systems for Advanced Applications (DASFAA 2001), April 2001, Hong Kong, IEEE Computer Society, pp.260-267
30. D. McCarthy and S. Sarin. Workflow and Transactions in InConcert. IEEE Data Engineering,16(2) (1993) 53-56
31. Ronald M. Lee. Documentary Petri Nets: A Modeling Representation for Electronic Trade Procedures. Business Process Management 2000: 359-375
32. Q. Li and F. H. Lochovsky. ADOME: an Advanced Object Modelling Environment. *IEEE Transactions on Knowledge and Data Engineering*, 10(2):255-276, 1998.
33. John A. Miller, Amit P. Sheth, Krys J. Kochut, and ZongWei Luo. 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 Nov. 1999.
34. Object Management Group. Foreword UML specification 1.4, September 2001.
35. R. G. Smith. The protocol net protocol: High Level Communication and Control in a Distributed Problem Solver, *IEEE Transactions on Computers* 29(12), December 1980, 1104-1113.
36. Staffware Corporation. Staffware Global - Staffware's Opportunity to Dominate Intranet based Workflow Automation, 2000, http://www.staffware.com
37. http://java.sun.com/products/ejb/
38. TIBCO Software Inc., which has acquired InConcert Inc., http://www.tibco.com

39. Workflow Management Coalition. Workflow Standard – Interoperability Wf-XML Binding, WFMC-TC-1023, May 2000.
40. Workflow Management Coalition. The Workflow Reference Model. (WFMC-TC-1003, 19-Jan-95, 1.1)
41. Workflow Management Coalition. Terminology and Glossay, WFMC-TC-1011, Feb 1999, 3.0.
42. Workflow Management Coalition - David Hollingsworth, ICL A&TC. White Paper – Events. April 1999.
43. http://www.w3.org/TR/wsdl