

# Views for Inter-organization Workflow in an E-commerce Environment

Dickson K.W. CHIU<sup>1</sup>, Kamalakar KARLAPALEM<sup>2</sup> AND Qing LI<sup>3</sup>

<sup>1</sup>*Dickson Computer Systems, 7A Victory Avenue, 4<sup>th</sup> floor, Homantin, Kowloon, Hong Kong:*

<sup>2</sup>*Indian Institute of Information Technology, Gachibowli, Hyderabad, INDIA:*

<sup>3</sup>*Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong*

*(email: kwchiu@dickson-computer.com, kamal@iiit.net, csqli@cityu.edu.hk)*

**Abstract:** In an e-commerce or 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 propose the use of workflow views as a fundamental support for workflow inter-operability and for visibility by external parties. We present taxonomy of workflow views, discuss various components of a workflow view, and their semantics with example usage. We also demonstrate how contracts can be facilitated with a supply-chain example.

**Keywords:** workflow management, workflow view, virtual enterprises, inter-organization workflow, e-commerce, meta-modeling, contracts, ECA rules.

## 1. INTRODUCTION

Workflow technology provides a way of modeling and automating business process. A Workflow Management System (WFMS) is a system that assists in defining, managing, and executing workflows. However, current research work mainly concentrates on workflows within a single organization. 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 inter-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 relevant and authorized 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 inter-organization workflow derivation. A workflow view can be regarded either as a structurally correct subset of a

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35658-7\\_21](https://doi.org/10.1007/978-0-387-35658-7_21)

R. Meersman et al. (eds.), *Semantic Issues in E-Commerce Systems*

© IFIP International Federation for Information Processing 2003

workflow definition (as in [Orlowska98]), 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).

To our best knowledge, there have not been attempts to explore views for workflows. A main contribution of this paper is thus to articulate the need and to introduce the concept of workflow views. Another contribution of this paper is to propose the use of workflow view as a fundamental support for inter-organizational workflow interoperability in an e-commerce environment, with a motivating example and taxonomy. The same workflow can interoperate with different workflows in different organizations without modification through different views. We also present a specification language for workflow views and highlight some implementation issues.

The rest of this paper is organized as follows. Sect. 2 presents a motivating example of a supply-chain e-commerce scenario. Sect. 3 presents taxonomy of workflow views and their operations. Sect. 4 details a specification language for workflow views. Sect. 5 highlights some implementation issues in our E-ADOME environment. Sect. 6 compares related work. Finally, we conclude this paper with further research issues in Sect 7.

## **2. A MOTIVATING EXAMPLE**

In this section, we present a motivating example of inter-organization workflow based on a supply chain e-commerce scenario, as depicted in *Figure 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.

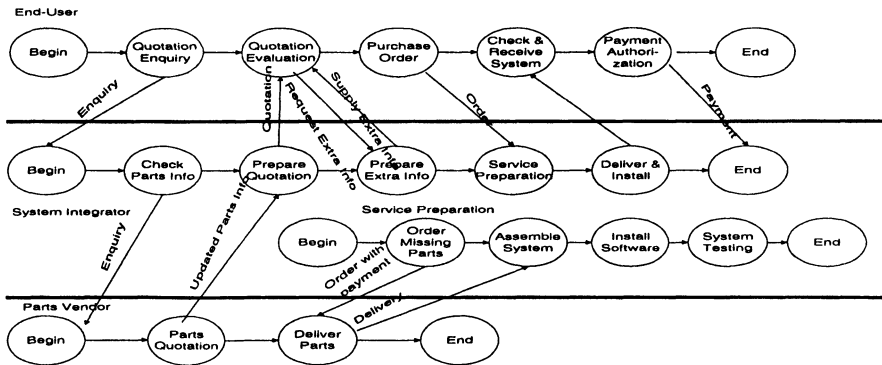


Figure 1. Inter-Organizational Workflow of a Supply-Chaining Example

## 2.1 Requirements of Workflow Views

In this example, the workflow view of the *end-user* presented to the system integrators has the following requirements. 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. Changes in delivery requirement or payment arrangement should notify the supplier. In particular, the enquiry process is concealed so that the system integrators can bid fairly and independently.

The workflow view of the *system integrator* presented to the end-user has the following requirements. The system integrator’s company profile and

technical specifications of the parts used in the proposed server are made available on request so that the end-user can evaluate the quotations more accurately. Changes in delivery date should be notified to the end-user. Further details of the process “service preparation” are available to the end-user so that the user can further monitor the progress of the job and estimate the delivery date. Updated quotation (price) is sent to the end-user upon a significant aggregated price change in parts with event-triggering mechanism during the evaluation process of the end-user.

The workflow view of the *parts vendors* presented to the system integrators has the following requirements. Updated price for the inquired parts is sent to the system integrators with event-triggering mechanism. Technical specifications and related information for the parts are made available upon request. Updates in software drivers and firmware will be notified to the system integrators using event-triggering mechanism (which in turn can notify the end-users). Changes in lead-time should be notified to the system integrators

## 2.2 Enforcing Contracts

With the various workflow views presented in the previous sub-section, the enforcement of the following contracts among different organizations can be facilitated. *Delivery schedule* - The system integrator can compute a delivery schedule to the end-user according to the vendor’s reported lead-time. If a vendor changes the lead-time, but the delivery schedule can still be completed within the end-user’s deadline, the change can be tolerated. Otherwise, another source for the part causing the problem is sought for, or an alternative part is used, subject to the end-user’s approval. *Price* - If the price for a certain part rises to an extent that there are no more profits, the system integrator may want to request an increase in price, use of an alternate cheaper part, 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. *Parts availability* - If a certain part is stopped from production, the system integrator may request the end-user’s approval of using an alternative part.

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

### 3. DIFFERENT KINDS OF WORKFLOW VIEWS AND THEIR OPERATIONS

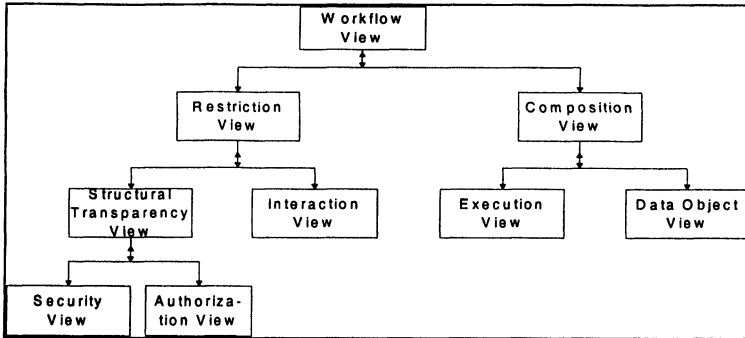


Figure 2. A Taxonomy of Workflow Views

Formally, based on Workflow Management Coalition (WfMC) workflow definitions [WfMC95], 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.

Motivated by the example application of Sect. 2, we present a taxonomy of workflow views in this section as an attempt to investigate this topic systematically. Our intention here is not to come up with an exhaustive taxonomy of workflow views, but to present some representative types of workflow views (as depicted below) to illustrate how they are useful in various situations relating to an e-commerce environment.

#### 3.1 Workflow Restriction Views

A workflow *restriction 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  $\forall (a, b) \in A'$ ,  $\exists n, \forall i, 1 \leq i \leq n, (t_i, t_{i+1}) \in A$  where  $t_1 = a$ ,  $t_n = b$  (the transitions in the view  $W'$  is based on any valid paths in  $W$ ). As shown in Fig.2, there are two specialized view types underneath: *Structural Transparency View* and *Interaction View*, the former in turn has two further subtypes, as discussed below.

A *Structural Transparency View* is for determining whether the internal details of a workflow are visible by a user, i.e., whether the workflow appears to be a black box (totally invisible), a white box (totally visible) or a grey box (partially visible). Since most contemporary WFMSs support hierarchical composition of tasks as workflows, the use of structural transparency can be facilitated. The workflow administrator can easily specify parts of a workflow to be visible, while other parts of a workflow not visible. Often, *security views* and *authorization views* are based on structural transparency views. For example, the requirement that the “Quotation Evaluation” process should be concealed while the “Service Preparation” should be transparent (cf. Sect. 138), is related to security and authorization to external business partners of the inter-organizational workflow. It should be noted that security view and authorization views are also useful to impose restrictions to members within an organization.

An *Interaction View* is the appearance of the workflow to an external user or agent to interact with the workflow. It is the subset of processes that has interaction with the user or agent. For example, the parts vendor’s interaction view with the system integrator consists of only the tasks “Check Parts Info” and “Order Missing Parts”.

### 3.2 Workflow Composition Views

A workflow composition view is a virtual workflow composed of components from different workflows, which may span across organizational boundaries. Formally, a workflow composition view  $W' = (T', A', J', F', X', V', E')$  is based on a set of workflows  $WS$  such that  $\forall W_i = (T_i, A_i, J_i, F_i, X_i, V_i, E_i) \in WS, T' \subseteq \cup T_i, J' \subseteq \cup J_i, F' \subseteq \cup F_i, X' \subseteq \cup X_i, V' \subseteq \cup V_i, E' \subseteq \cup E_i$ , and  $\forall (a, b) \in A', (\exists n, \forall i, 1 \leq i \leq n, (t_i, t_{i+1}) \in \cup A_i$  where  $t_1=a, t_n=b$ ) or  $(a \in A_1, b \in A_2, A_1 \neq A_2$  and  $A_1, A_2 \in \cup A_i)$  (the transitions in the view  $W'$  is based on any valid paths in  $W$  or transitions from one workflow to another workflow).

*Figure 3* illustrates a composed view of the supply-chaining motivating example, which is composed from the three workflows in three different organizations. In particular, this is also an execution view of a cross-organizational workflow. An *Execution View* covers the set of processes that constitute one or more typical execution sequences under a target scenario. A *Data Object View* is the set of tasks in which the target data / object (such as forms) are accessed. Typical examples are purchase request forms, invoices, purchase order, etc. Data object view may also span across organizational boundaries (e.g., purchase order).

### 3.3 Discussion

The above taxonomy of views for workflow has been proposed within the context of inter-organizational workflows. In general, the taxonomy for workflow views depends on the usage of these views. In that regard, this taxonomy provides a preliminary classification based on well-defined concepts of restricted and composition views. A composition view conceptualizes an aggregation of a set of workflows as a higher-level workflow, whereas a restricted view conceptualizes a part of a workflow. There is scope for developing this taxonomy further depending on the usage and utility of views for specific applications. The main advantages of workflow views are: ease of specifying them, clarity of their meaning, and ease of implementing view support in a WFMS. In rest of the paper, we digress on these aspects.

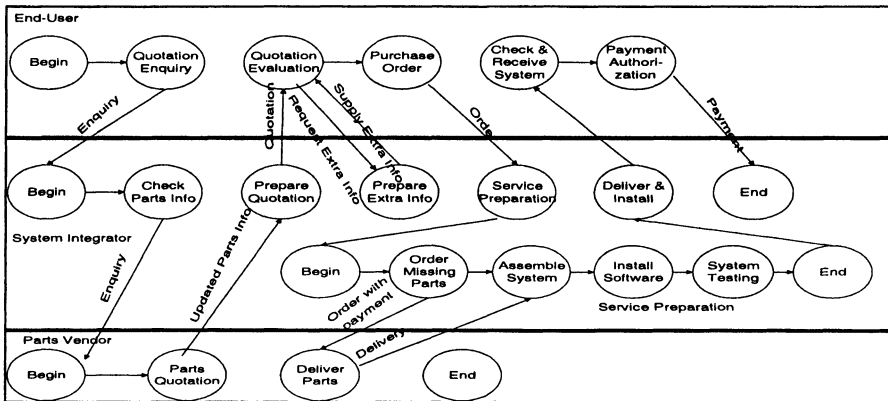


Figure 3. Inter-Organizational Composed Workflow View of Supply-Chaining

### 3.4 Operations Required for Supporting Workflow Views

The following operations are required for supporting workflow views:

- Processing of workflow view definitions, i.e., translating the workflow definition language to an internal representation;
- Creation of a view for a workflow instance when it is instantiated or when another external workflow is invoked;
- Presentation of the view definition upon user's request, i.e. the meta-data of the workflow made available to the user;

- Processing of user's request for the content/update of objects, process flow diagram, process progress (i.e., the current sub-process in execution), with verification of access;
- Definition of user specified events/exceptions based on the information accessible to them;
- Send appropriate messages to the user upon events and exceptions;
- Update the view definition upon workflow evolution to maintain consistency, e.g., removing objects and attributes that are deleted during workflow evolution.

### 3.5 View Definition Update Policies

When workflow views are updated, there can be different policies regarding to the current users of the affected view: *Temporary change* only changes a certain instance (e.g., as a trial). *Delayed change* changes future instances only, the other instances currently executing continue using the old definition. *Immediate change* changes all the instances suddenly use the new definition (therefore the user should judge whether this has inconsistency impact). *Progressive change* allows different instances to take different decisions on whether to change over to the new view.

## 4. COMPONENTS OF A WORKFLOW VIEW

Based on the formal definition in Sect. 3, 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 all components are modeled as objects in most advanced object WFMS, we discuss them separately because each of them has different semantics. *Figure 4* depicts a simple workflow view meta-class definition based on the ADOME-WFMS workflow meta-definition ([ChiuLiKa99]).

```

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, ...}
end

```

*Figure 4.* Workflow View Meta-class Definition



## 4.1 Process Flow Graph

Most contemporary WFMSs use a hierarchical composition approach, i.e., a process (workflow or activity) 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 detailed 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 (e.g., "Service Preparation" in *Figure 1*) is specified with a sub-workflow view by a statement "process p1 view v1", i.e., the details of the sub-process are further visible and subject to the restriction of a sub-workflow view. A *black-box* sub-process (e.g., "Quotation Enquiry" in *Figure 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 *grey-box* is 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.

## 4.2 Objects associated with a workflow instance

An object associated with a workflow instance need not be exposed 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 being available only upon user's request.

The "object" statement in *Figure 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 is 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.

### **4.3 Events and exceptions**

When events and exceptions are specified in a view, a mechanism, such as a corresponding message, should notify the view user upon their occurrences. This is particularly useful in providing inter-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, users need not poll on their interested objects and thus increase the efficiency. For example, an 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 than expected.

### **4.4 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.

### **4.5 Discussion**

A workflow is composed of many inter-related objects. Therefore, a workflow view is a sub-workflow that adheres to these inter-relationships. In order to achieve this, the view definition presented has to have constructs for referring to these inter-relationships. In the workflow view taxonomy, we

have not classified the views based on these inter-relationships, as these inter-relationships can be exploited in both restricted and composition views. In case there are no explicit restrictions limiting the interrelationships, all the inter-relationships will be included in the definition of the view by default. One can always define object-based views that are based on these interrelationships within a workflow.

## 5. IMPLEMENTATION ISSUES

```

class View
  W: Process;
  Process_view: set of (Process, View);
  Process_rename: set of (Name, Process);
  Transition_rename: set of (Name, Process, Process);
  Object_view: set of Object_def;
  Attribute_view: set of Attribute_def;
  Event_view, Exception_view: set of (name, event);
  Rule_view: set of (name, event),
end
class Object_def
  Object_name: Object;
  Object_expression: Expression;
  Write, Input, Output: Boolean;
end
class Attribute_def
  Attribute_name: Attribute;
  Write, Read, Denied: Boolean;
end

```

Figure 5. Workflow View Class Definition

In this section, we briefly look at some implementation issues related to workflow view mechanism. Our discussions mainly focus on E-ADOME [ChiuKaLi00], an extension of ADOME-WFMS that we have developed for supporting cooperative handling of workflow exceptions [ChiuLiKa99, ChiuLiKa01]. If the underlying object-base of the WFMS does not support derived objects, we must explicitly store view definitions in a class. The required class definitions are as shown in *Figure 5*.

As shown in *Figure 6*, 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 [LiLoch98], to allow them to more adequately deal with data and knowledge management requirements of advanced information management applications, especially WFMSs. The ADOME prototype has been built by integrating an OODBMS (ITASCA [IbexInc94]) and production inference engine (CLIPS [CLIPS92]). Therefore, a WFMS can be implemented on top of it with relative ease. *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 [Chiu00, ChiuLiKa99]. *E-ADOME Interface Layer* – this is the

enhancement layer to the WFMS so as to enable ADOME-WFMS to interact with various types of external agents through the Internet. *E-service Agents* – these are agents providing services or solving problems, and are external to the E-ADOME system.

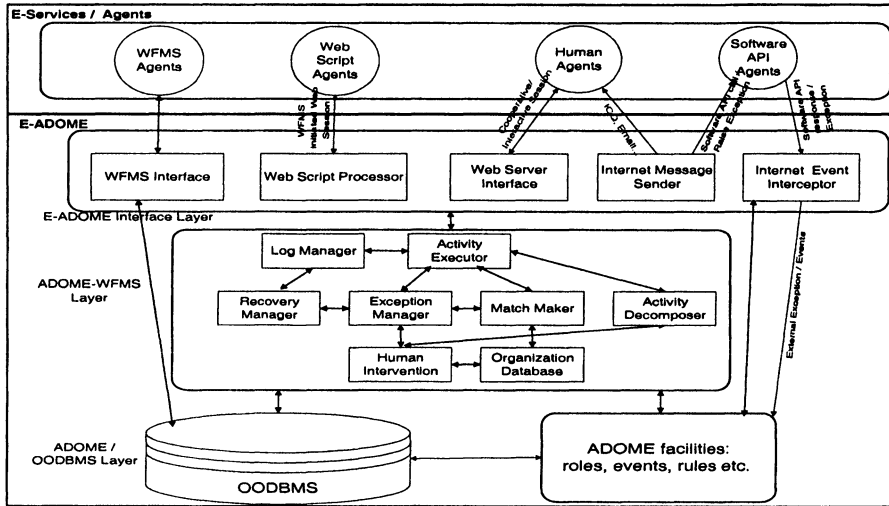


Figure 6. E-ADOME Environment

Currently, we are enhancing E-ADOME for e-service and e-commercial inter-organizational workflow applications, particularly the view mechanism as outlined in Sect. 3 and 4. We are adding the feature of workflow views to E-ADOME in the “WFMS Interface” module for inter-organization workflow to enable more interactions with external WFMS agents (such as the system integrator workflow depicted in *Figure 6*).

## 6. 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 [Dayal89], and in particular by imaginary objects in [Abiteboul91]. [Gardarin97] discusses federated OODBMS and views for objects in a distributed environment. On the other hand, it is a relatively new approach to E-commerce / E-service enactment based on an advanced WFMS engine. Besides E-ADOME [ChiuKaLi00], other notable systems using related approaches include Eflow [Casati00] and Crossflow [Grefen98].

As for standards, Workflow Management Coalition (WfMC) has recently proposed an Interworkflow Application Model to model inter-organizational workflow and Wf-XML [WfMC00], which is an interchange format specification for an XML language designed to model the data transfer requirements for inter-operating process specification. These documents describe the modelling and specification of the overall workflows and inter-organizational messages. However, they do not provide information-hiding mechanisms for providing only partial visibility of processes to other organizations. Our approach also allows each organization to advertise its workflow views (i.e., its relevant parts of workflow and other information) to the public so that prospective partners may use interfacing techniques or adapt their own workflows to inter-operate. This approach is motivated by the concept that organizations are agents running upon WFMSs [ChiuKaLi00]. Unless there can be standardisation of workflows up to the application layer for individual trades (which is unlikely due to the diversity of the natural of business entities), our approach provides the most opportunities for interoperation.

Crossflow [Grefen98] 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). High-level support is obtained by abstracting services and offering advanced cooperation support. Virtual organizations are dynamically formed by contract-based matchmaking between service providers and consumers. However, contracts are complementary to workflow views because workflow views offer enhanced visibility, help contract negotiation and enactment. Moreover, workflow views can also facilitate B-to-C E-commerce or E-service, in which an end-user participates in an organization's workflow.

WW-flow [Kim00] provides a hierarchical control scheme over activities 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, but does not address issues across organization boundaries. Eflow [Casati00] 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. Dartflow [Cai96] is one of the first web-based WFMS, using transportable agents, CGI and Java technologies. Currently, several commercial WFMSs such as TIB/InConcert [TIBCO00], Staffware 2000 [Staffware00], provide web user interface too. In addition, I-Flow [Enix00] has a Java workflow engine. However, these products or prototypes do not address the issue of inter-organizational workflows at the date of this paper.

## 7. CONCLUSIONS

This paper has presented a novel concept of workflow view, and its possible applications in an e-commerce environment. With improved visibility through workflow views, it is perceived that contract negotiation and enforcement can be facilitated [KafChiu01]. Through a practical motivating example of supply chain, we presented a brief taxonomy of workflow views with respect to that particular application environment. We have also presented a meta-class definition for workflow views, and illustrated their detailed structures and possible semantics. Furthermore, we highlighted some implementation issues for workflow views. It is perceived that workflow views can be easily implemented in most WFMS. Currently, we are adding workflow views in the interface layer of our E-ADOME advanced workflow environment for inter-organization workflows.

We are investigating further details of the application of workflow views in contract negotiation and management, security issues, for B-to-C applications, and any other application possibilities. We consider further research issues on interfacing and interoperability important for extending the applicability of an advanced WFMS engine, which includes: expanding the possible interfaces and coordinating different types of agents, graphical workflow evolution tools, and inter-operating with other WFMS. We are interested in the application of ADOME-WFMS in various advanced real-life e-commerce environments, such as procurement, supply-chain, finance, stock trading and insurance. On the other hand, we are working on a more complete taxonomy of workflow views, integrity / semantic constraints for workflow views, and investigating its usage in a wider range of applications.

## 8. REFERENCES

- [Abiteboul91] S. Abiteboul, A. Bonner. Objects and Views. In Proceedings of ACM SIGMOD Conference, 1991.
- [Berry99] K.L. Myers, and P.M. Berry. At the Boundary of Workflow and AI. In proceedings of the AAAI-99 Workshop on Agent-Based Systems in The Business Context held as part of AAAI-99, Orlando, Florida, July 1999.
- [Cai96] 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.
- [Casati00] F. Casati, et al. Adaptive and Dynamic Service Composition in eFlow. HP Laboratories Technical Report HPL-2000-39, March 2000.
- [ChiuKaLi00] 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, Egypt, Sept. 2000.

- [ChiuLiKa99] D.K.W. Chiu, Q. Li and K. Karlapalem,. A Meta Modeling Approach for Workflow Management System Supporting Exception Handling. Special Issue on Method Engineering and Metamodeling, *Information Systems*, Pergamon Press, Elsevier Science, 24(2):159-184, 1999.
- [ChiuLiKa01] D.K.W. Chiu, Q. Li and K. Karlapalem. Web Interface-Driven Cooperative Exception Handling in ADOME Workflow Management System. *Information Systems*, Pergamon Press, Elsevier Science, 26(2):93-120, 2001.
- [CLIPS] GHG Corp. Clips Architecture Manual, Version 5.1. Jan 1992. (available at <http://www.ghg.net/clips/CLIPS.html>)
- [Dayal89] U. Dayal. Queries and Views in an Object-Oriented Data Model. In Proceedings of 2<sup>nd</sup> International Workshop on Database Programming Languages, 1989.
- [KafChiu01] E. Kafeza, D.K.W. Chiu and I. Kafeza. View-based Contracts in an E-service Cross-Organizational Workflow Environment, 2nd VLDB Workshop on Technologies for E-Services, Rome, Italy, Sept 2001, pp.74-78, LNCS 2193, Springer.
- [Enix00] Enix Consulting Limited. An Independent Evaluation of i-Flow Version 3.5, 2000 (available at <http://www.i-flow.com>).
- [Gardarin97] G. Gardarin, B. Finance, 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.
- [Hoffner00] Y. Hoffner, H. Ludwig, C. Gülcü, P. Grefen; Architecture for Cross-Organisational Business Processes. In *Proceedings of the Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000)*, pp. 2-11, Milpitas, CA, June 8 and 9, 2000. IEEE Computer Society, Los Alamitos.
- [HP98] Hewlett Packard. Changengine Admin Edition Process Design Guide, 1998.
- [IbexInc94] Itasca Reference Manual, Ibex Corporation, 1994.
- [ICQ00] <http://www.icq.com>
- [Kim00] 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.
- [LiLoch98] Q. Li and F. H. Lochovsky. ADOME: an Advanced Object Modeling Environment. *IEEE Transactions on Knowledge and Data Engineering*, 10(2):255-276, 1998.
- [Orlowska98] A. ter Hofstede, M. Orlowska, and J. Rajapakse. Verification Problems in Conceptual Workflow Specifications. *Data & Knowledge Engineering*, Pergamon Press, Elsevier Science, 24(3):239-256, Jan 1998.
- [Staffware00] Staffware Corporation. Staffware Global - Staffware's Opportunity to Dominate Intranet based Workflow Automation, 2000, <http://www.staffware.com>
- [Sycara96] K. Sycara, and D. Zeng. Coordination of Multiple Intelligent Software Agents, *International Journal of Cooperative Information Systems*, Vol. 5, Nos. 2 & 3, 1996.
- [TIBCO00] TIBCO Software Inc., which has acquired InConcert Inc., <http://www.tibco.com>
- [WfMC95] Workflow Management Coalition. The Workflow Reference Model (WFMC-TC-1003, 19-Jan-95, 1.1)
- [WfMC98] Interworkflow Application Model: The Design of Cross-Organizational Workflow Processes and Distributed Operations Management
- [WfMC00] Workflow Management Coalition. Workflow Standard – Interoperability Wf-XML Binding, WFMC-TC-1023, May 2000.

# **Session IV: Semantics Mining**