

THREE-TIER VIEW-BASED MOBILE WORKFLOWS FOR MULTIPLE PLATFORMS

Dickson K.W. CHIU¹, Shing-Chi CHEUNG² and Eleanna KAFEZA²

¹Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong

²Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong
email: kwchiu@acm.org, {scc, kafeza}@cs.ust.hk

ABSTRACT

Workflow technology has recently been employed not only within businesses but also as a framework for implementing services over the Internet. Recent advances in mobile technologies have led to increasing demands for mobile accesses to workflow management systems (WFMS). The basic requirement is to support SMS, WAP and web browsers on PDAs, in addition to regular web browsers on PCs. As the capabilities and bandwidth of these mobile devices are significantly inferior to computers over regular Internet connections, workflows have to be adapted to accommodate these limitations. Instead of redesigning or adapting workflows in an ad-hoc manner for different kinds of platforms, we propose a framework of workflow adaptation for mobile users based on three tiers of views: user interface views, data views and workflow views. User interface views provide alternative presentations of inputs and outputs. Data views summarize data over limited bandwidth and display them in different forms. Furthermore, we introduce a novel approach of applying workflow views to mobile workflow adaptation, where mobile users may execute a more concise version or modified procedures of a business process. The workflow view also serves as the centric mechanism for integrating user interface views and data views. We demonstrate the feasibility of our approach by extending a flexible web-based WFMS E-ADOME into ME-ADOME. The concepts are illustrated with a workflow of procurement approval.

1. INTRODUCTION

Workflow is the computerized facilitation or automation of a business process. A business process is a set of interdependent 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. Traditional WFMSs often can only coordinate workflows and their enacting agents (often limited to software processes) within a single organization. However, contemporary WFMSs (such as [3], [4], [16], [23], [27], [29]) can now interact with various types of distributed users and agents over the Internet. In this article, the term workflow is used to refer to this more general notion of process management. With recent advances in mobile technologies and the availability of telecommunication infrastructures [24], there are increasing demands for mobile accesses to WFMSs, especially because many organizations may have already employed workflow management in their business processes. Although existing WFMSs are mostly web-enabled, there are few studies on the exploration of mobile workflow support ([22], [28], [30]).

We have done some preliminary work [5] 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 services over the Internet for agents when delegated to by the users), and the system for

supporting them efficiently, with all the desirable features provided by the underlying WFMS. Furthermore, we have proposed a novel approach of applying *workflow view* in [5] for e-contract enactment in a cross-organizational workflow environment. As follow-up work, we detail in [5] how mobile workflows can be facilitated by *views*, with respect to our E-ADOME workflow engine extended with mobile support, i.e., ME-ADOME. Since a workflow could involve different users on heterogeneous platforms, different views can be adapted to the needs of these users based on various platform requirements. Thus, we can add mobile support to workflows in a manageable manner, within a relative short period of time for the adaptation. In this article, we further discuss some guidelines for designing these views.

The contribution and coverage of this article are as follows: (i) a three-tier view-based framework for supporting mobile workflow, (ii) a novel approach of applying workflow views in supporting workflow adaptation in a mobile workflow environment, (iii) separation of user alert and user session for flexible workflow enactment, (iv) details on the technologies for supporting workflows on mobile platforms, (v) demonstration of the applicability of ME-ADOME in supporting mobile workflow through these features, and (vi) some guideline for designing these views.

The rest of our article is organized as follows. Section 2 presents a motivating example and the limitations of mobile platforms. Section 3 presents our three-tier view-based framework for supporting mobile workflows. Section 4 presents our ME-ADOME architecture to illustrate how a flexible WFMS engine can be extended to coordinate mobile users on different platforms, with various view support. Section 5 discusses some guidelines for designing views based on our experiences. Section 6 compares related work. Finally, we conclude this article with our plans for further research in Section 7.

2. MOTIVATION AND BACKGROUND

In this section, we first present a motivating example of procurement approval within a purchase request workflow, and then discuss an overview of the limitations of current mobile platforms.

2.1 Motivation Example

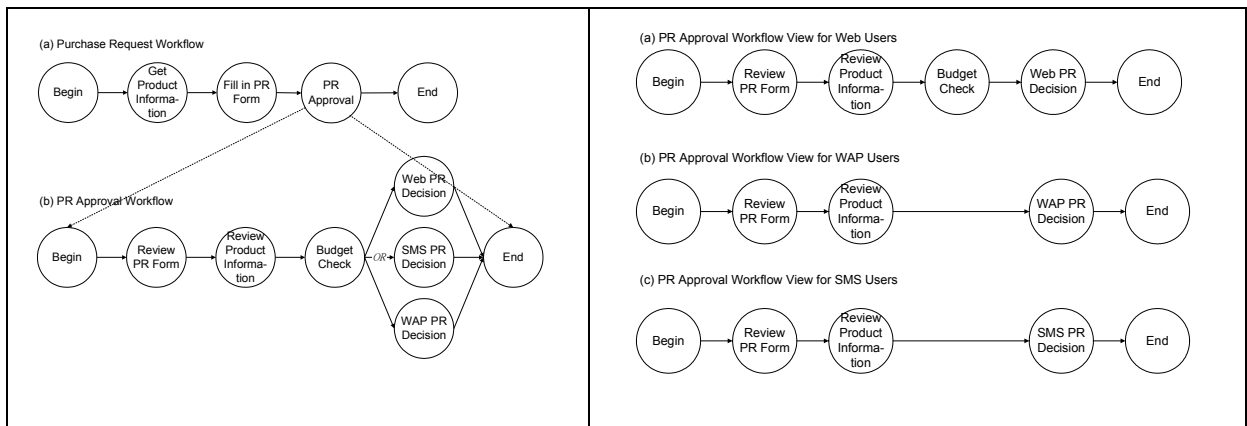


Figure 1: Example approval workflow for purchase request and different views

As depicted in Figure 1(a), in the *purchase request (PR)* workflow, an employee first collects the information of the required product and inputs the information into the WFMS. Then he/she fills in a PR form through the WFMS. The most critical step is the approval procedure, which is to be carried out by an authorized officer, such as a department manager. This step in the workflow is delayed when the authorized officer has put aside the form, or is not at office for the decision. If the authorized officer is on leave or on a business trip, this may cause weeks of delay. Even if the officer

is away for several hours, the delay may cause catastrophic effects to critical projects, especially if other parts of the workflow (such as B-to-B interactions) have been automated [5].

As mobile technologies [24] become mature and generally available, the WFMS may alert the officer for immediate decision through mobile devices. The alert, for example, can be sent via *Short Message Services* (SMS) over mobile phones. Then the officer may connect to WFMS with the mobile phone via *Wireless Application Protocol* (WAP) or simply reply with SMS messages. Alternatively, if further information is required, the officer may access the WFMS before making decision with a *Personal Digital Assistant* (PDA) over a mobile network or a computer in a net cafe. Such extensions to our previous approach [8] with user alerts by means of ICQ messages [20] or electronic mails, facilitates timely decision making while users are away from office.

However, supporting mobile devices for WFMS is a challenging task. There are severe hardware limitations such as processing power, memory, disk space, screen size, input ergonomics and communication bandwidth. Therefore, workflows have to be adapted to accommodate these limitations. Instead of redesigning or adapting workflows in an ad-hoc manner for different kinds of platforms, we propose a framework of workflow adaptation for mobile users based on three tiers of views: user interface views, data views and workflow views. Figure 1(b) further explains the details of the *purchase request approval* workflow. The authorized officer first reviews the PR form and the product information, and then checks whether the budget is available. Finally, the officer makes the decision. For example, the last decision step is customized into three alternatives, viz. *Web PR decision*, *SMS PR decision* and *WAP PR decision*. However, this approach is quite ad-hoc and cannot describe customization in other activities of this workflow in an orderly manner.

Therefore, we propose to address this problem using views. Figure 1(c)-(e) illustrate three different workflow views of the *purchase request approval* workflow for users on three different platforms, viz., standard web browser, WAP and SMS respectively. For WAP and SMS users, the budget check is skipped because it may be either delegated to other colleagues or postponed. In addition, each type of users can execute a variant of the *PR decision* activity. As a result, the overall customization of different workflows can be orderly managed by means of views.

It should be noted that the activities *review PR form* and *review product information* are not customized at the workflow tier, but at the *user interface tier* instead, i.e., just the appearance of the screens and input is different while they are referring to the same set of information items. This can be facilitated by contemporary XML technologies augmented with XSL in elegantly, as further explained in the next section. Similarly, users on PDA-based web browsers require just different user interface views to cater for the small screen size. Figure 2 illustrates some user interface views supporting WAP users within the *purchase request approval* workflow. Similarly, users on PDA-based web browsers require just different user interface views to cater for the small screen size.

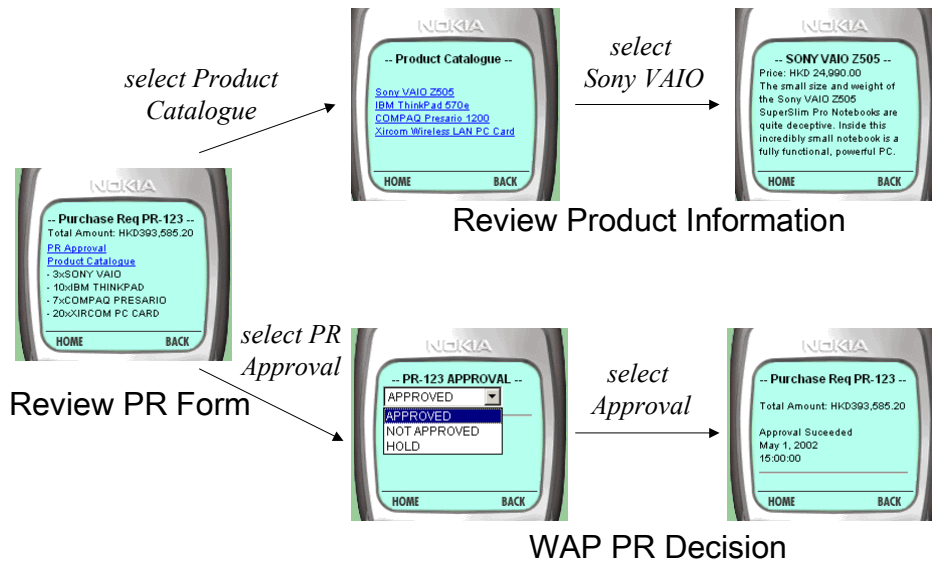


Figure 2: User Interface Views Supporting the WAP Based Workflow View in Figure 1(d)

2.2 Technological Limitations of Mobile Platforms

In this sub-section, we present an overview of the technological limitations of different mobile platforms, viz. PDA, WAP and SMS, with reference to mobile workflow requirements. Figure 3 summarizes the contribution of different types of views to users on these mobile platforms respectively.

Platforms \ Views	PDA	WAP	SMS
User Interface View	Simplified screen layout Low resolution graphics Panning and Zooming	WML translation Highly reduced screen	SMS message presentation
Data View	Omit some fields Summarized information	Mandatory fields only Highly summarized information	Highly summarized and mandatory information as message content
Workflow View	Simplified workflow steps and procedures		SMS dialogue presentation

Figure 3: Contribution of Different Views to Users on Various Platforms

2.2.1 Short Message Services

The Short Message Service (SMS) supports sending and receiving text messages to and from mobile telephones. The text can comprise words or numbers or an alphanumeric combination. SMS was created when it was incorporated into the Global System for Mobiles (GSM) digital mobile phone standard. A single short message can be up to 160 characters of text in length using default GSM alphabet coding, 140 characters when Cyrillic character set is used and 70 characters when UCS2 international character coding is used.

The advantage of SMS is its high availability. All major mobile networks worldwide and most handsets now support SMS. In mobile workflow applications, it is useful for notifying the user who is

to be involved in a workflow activity, for example, with a short summary message or codify message (e.g., "Open Sesame"). In addition, the WFMS can also send a URL to the user, so that the user can go to a better device (e.g., a PC in a net cafe) and log onto the system through a regular browser. If the user has already learnt the details of the approval request beforehand or from colleagues over the phone, he can even send in his decision by replying the system with a SMS message.

However, the main limitation of SMS is its short message length. Next generation Extended Message Service (EMS), which is available in some newer handset models, definitely helps as it supports voice, graphics, and longer messages. Sometimes, mobile users may have their phones switched off or disconnected from phone networks. In addition, it is difficult to write reply messages with mobile handsets. Security and authentication may also be a problem in SMS.

2.2.2 Wireless Application Protocol

Wireless Application Protocol (WAP) is an open standard for application that runs over mobile networks. The standard allows the display of web contents and the execution of Internet applications on wireless devices. WAP applications are developed in *Wireless Markup Language* (WML), based on *eXtensible Markup Language* (XML). Decks/Cards are used by WML to specify a service. A card is a unit of interaction with which the user is either displaying or requesting information from applications or other users. A collection of cards is known as a *deck* that constitutes the services. This approach is useful as inter-page navigation can be avoided. As such, interactive sessions with the WFMS can be supported. Users can read the provided summary information, and/or email before making decision. If necessary, users may request to expand selected summaries into original documents.

However, the main problem with WAP phones is their tiny screens though it may be improved with better handset. This nevertheless limits the amount of information that can be displayed. It is also difficult to write reply messages with a handset. WAP sessions may be aborted because of poor network connections and WAP gateway failures, and are therefore less robust than SMS messaging.

2.2.3 Personal Digital Assistants

Personal Digital Assistants (PDAs) once connected to the Internet, via modems or wireless local area networks (LAN), can support user to carry out interactive sessions with a WFMS through a browser. PDA screens are smaller than PC (and some cannot display colors), but are still quite informative. PDA users can also read full email messages and participate in ICQ communications. Many PDAs support native code applications and/or Java applets, so that specialized workflow client applications can also be supported. Input facilities for PDA (e.g., via pen or keyboard) are much more convenient than mobile phone handsets.

The main limitation of PDAs in comparison with desktop PCs is that they cannot display complex web pages, high-resolution graphics or large documents. Moreover, PDAs usually cannot operate too long (for just a few hours) because of their battery capacity, and therefore cannot support constant user connectivity for alerts like mobile phones. However, these limitations are gradually overcome by improved PDA hardware and better data rate of third-generation (3G) mobile networks. Moreover, the WFMS may first alert a user with SMS, and then the user connects to the WFMS via a PDA.

3. A THREE-TIER VIEW-BASED SUPPORT MODEL

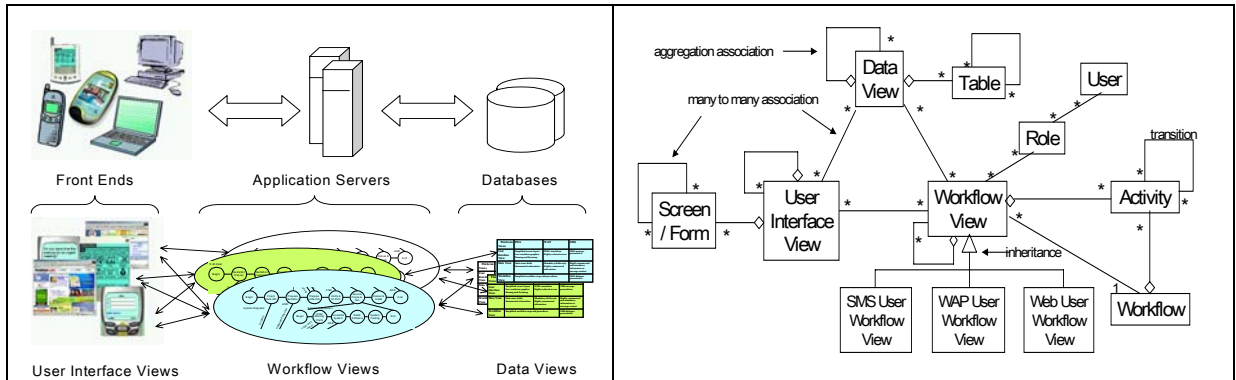


Figure 4: A meta-model of three-tier views (in UML) and their relations between three-tier architecture

As discussed in the previous section, in general, the capabilities and bandwidth of mobile devices are inferior to desktop computers over wired Internet connections. Therefore, workflows have to be adapted to accommodate these limitations. Instead of redesigning or adapting workflows in an ad-hoc manner for different kinds of platforms, we propose a framework of workflow adaptation for mobile users based on three tiers of views. Internet applications are generally developed with a three-tier architecture comprising front ends, application servers and back end databases. Each of these tiers hosts a set of views as shown in Figure 4.

At the front end tier, user interface views provide mobile users with alternative presentations and interfaces to interact with the workflow views hosted at the application servers tier. User interface views can be developed based on the technology of XML Style Language (XSL) [33]. A user interface view consists of multiple screens and forms based on the capabilities of the front end devices. For example, Figure 6(a) and (b) give the screens in two user interface views for devices with and without graphical capabilities, respectively. A workflow view is a projection of workflow activities that concern a user and his/her front end device. Workflow views are synchronized and enacted by application servers. A data view consists of multiple tables that collectively represent a projection of data that are required in the enactment of a work view.

Figure 4 also presents our meta-model of the three-tier views in the Unified Modeling Language (UML [27]), which is a modeling language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components. User interface, workflow and data views are related by many to many relations. In other words, a user interface view may provide the interface to multiple workflow views, each of which may in turn supports multiple user interface views. Views help balance trust and security, that is, only information necessary for the process enactment, enforcement and monitoring of the contract is made available to the concerned parties, in a fully control and understandable manner.

3.1 Workflow Views

Motivated by views in federated object databases, we propose the use of workflow views as a fundamental mechanism for flexible workflow adaptation. A workflow view is a structurally correct subset of a workflow definition (as defined in [16]). We propose to use the concept of workflow views to let different users (on different platforms) access different customized version of the same workflow. Within an organization, workflow views are also useful for security applications, such as to restrict accesses (like the use of views in databases). Thus, workflow views serve as the centric mechanism in our approach. They represent customized business processes that integrate with data

views and user interface views, as illustrated with our motivating example in the previous section. Based on our meta-model presented in Figure 4, we develop a simple workflow view definition language in accordance as depicted in Figure 5. Figure 5 also lists the specification of two example workflow views of the PR approval process for WAP users (cf. Figure 1(d)) and SMS users (cf. Figure 1(e)) respectively, each containing a customized version of the workflow, user interfaces and data views. The components of a workflow view include the customized process flow graph, access control, related data views and user interface views.

Access (Security) Control - Each workflow view must be specified with one or more accessible *roles* with an *access* statement. A role represents a collection of users of similar properties [7]. While some roles (e.g., SMS, WAP, Web) are used to distinguish user platforms for mobile workflow support, other roles (e.g. director, manager, officer) may also be used in specifying security context.

Process Flow Graph - Most contemporary WFMSs use a hierarchical composition approach, i.e., a workflow (process) is composed of sub-workflows and so on down to leaf-nodes of atomic activities. 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-workflow may be concealed. Thus a process in the flow graph can be presented in one of the following ways: (i) a white-box sub-workflow is specified with a sub-workflow view by a statement "process p1 view v1" (i.e., the details of the sub-workflow is further visible and subject to the restriction of a sub-workflow view); (ii) a black-box sub-workflow (is limited from further details of its further internal composition; (iii) a gray box where some sub-workflows are visible while other sub-workflows are concealed. Similarly, we can define customized parts the process flow graph for different parts of a composite workflow.

Data views associated with a workflow instance – Data tables associated with a workflow instance need not be presented completely in a workflow view. Some fields can be hidden from the views, some can be read only, while some are presented with write access. Data views are specified with the *dataview* statement. We omit further details of data views as they can be specified with standard Structure Query Language (SQL) statements [12]. Further examples will be given in following sub-sections.

User interface views associated with a workflow instance – User interface views (further specified in XSL) associated with a workflow instance specify different user interfaces for users on different platforms. Further details and examples are given in the next sub-section.

<pre>view v of workflow w begin {access role1, role2, ... } {process p1 view v1, ...} {transition p4 to p5, ...} {dataview dv1, dv2, ... } {UIview UIv1, UIv2, ... } end</pre>	<pre>view WAP_user of workflow PR_Approval begin access WAP; process Review_PR_Form, Review_Product_Information, WAP_PR_Decision; transition BEGIN to Review_PR_form, Review_PR_form to Review_Product_Information, Review_Product_Information to WAP_PR_Decision, WAP_PR_Decision to END; dataview WAP_data_view; UIview WAP_UI_view; ... end view SMS_user of workflow PR_Approval begin access SMS; process Review_PR_Form, Review_Product_Information, SMS_PR_Decision; transition BEGIN to Review_PR_form, Review_PR_form to Review_Product_Information, Review_Product_Information to SMS_PR_Decision, SMS_PR_Decision to END; dataview SMS_data_view; UIview SMS_UI_view; ... end</pre>
--	---

Figure 5: Workflow view definition language and example

3.2 User Interface Views

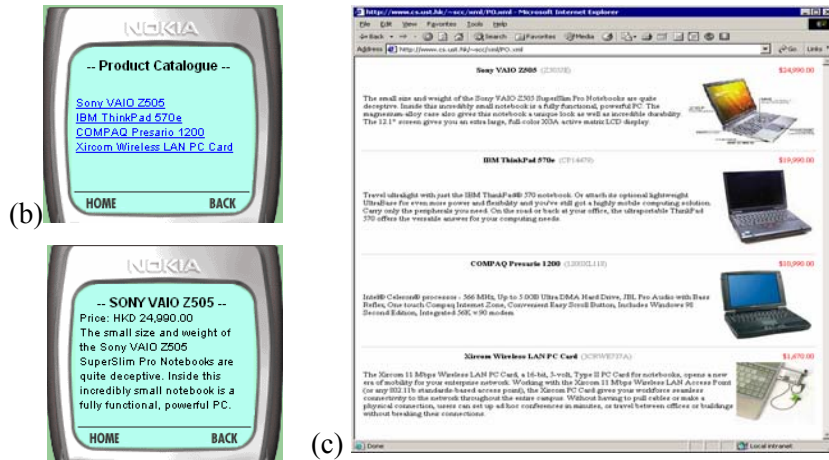


Figure 6: Examples screens for (a) WAP users and (b) web users

User interface views provide users with appropriate interfaces to interact with workflow views within the capabilities of front end devices. This means the user interface views for a web user can be different from that for a WAP user. Figure 6 presents two possible user interface views that support the *Review Product Information* activity in the workflow views depicted in Figure 5 for Web users and WAP users, respectively. Due to device limitations, multimedia contents, such as pictures, animations and videos, are omitted from the screens in the user interface view for WAP users. Both Figure 6(b) and (c) correspond to the same XML document object but are rendered by two different XSL style sheets. Figure 6(b) gives the product description of SONY VAIO Z595 after a user clicks the first hyperlink in the product catalogue in the left screen.

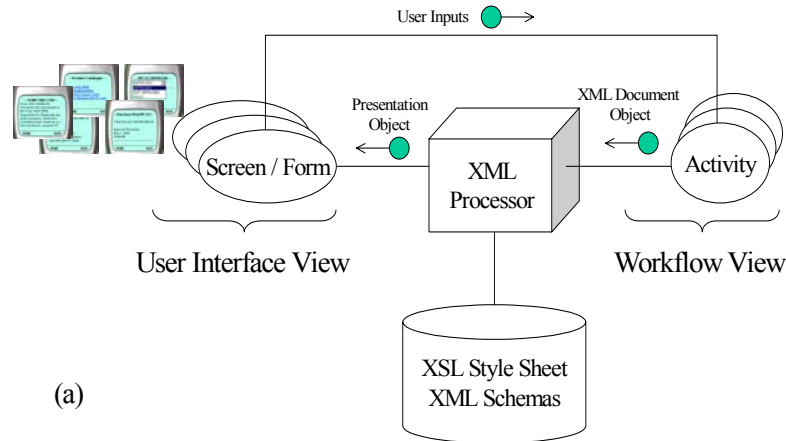


Figure 7: Implementation framework for user interface view

Figure 7 presents an implementation framework for user interface views using XSL technology [33]. Information to be presented at a user interface view is structured as XML document objects by a workflow view, such as the WAP based workflow view Figure 6(b). Figure 8 gives an XML document object of a catalogue generated by the *Review Production Information* activity. XML document objects are then transformed by an XML processor into presentation objects based on specified XML schemas and XSL style sheets. For example, the presentation objects for a WAP user interface view are decks and cards in the Wireless Markup Language (WML). A sample presentation object for the *Review PR Form* supporting the WAP based workflow is shown in Figure 9.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- <!DOCTYPE XMI SYSTEM "CatML.dtd" -->
<XMI xmi.version="1.1" timestamp="TUE Apr 30 2002" >
  <XMI.header>
    <XMI.metamodel xmi.name="CatML" xmi.version="1.0"/>
  </XMI.header>
  <XMI.content>
    <Product xmi.id="sku-Z505JE">
      <CatalogItem.name>Sony VAIO Z505</CatalogItem.name>
      <CatalogItem.description>
        The small size and weight of the Sony VAIO Z505 SuperSlim Pro Notebooks are quite
        deceptive. Inside this incredibly small notebook is a fully functional, powerful PC.
        The magnesium-alloy case also gives this notebook a unique look as well as
        incredible durability. The 12.1" screen gives you an extra large, full-color XGA
        active matrix LCD display.
      </CatalogItem.description>
      <CatalogItem.listPrice>
        <Money>
          <Money.currency>HKD</Money.currency>
          <Money.amount>24,990.00</Money.amount>
        </Money>
      </CatalogItem.listPrice>
      <CatalogItem.sku>Z505JE</CatalogItem.sku>
      <CatalogItem.category>
        <Category href="Taxonomy.xml#Laptop_Computer_System" xmi.label="Laptop Computer
        System"/>
      </CatalogItem.category>
      <CatalogItem.detail>
        <Resource href="http://www.dickson.com/products/sony/z505/" xmi.label="Product
        Specifications"/>
      </CatalogItem.detail>
      <CatalogItem.supplier>
        <Party href="Suppliers.xml#Dickson" xmi.label="Dickson Computer Ltd."/>
      </CatalogItem.supplier>
      <Product.photoURL>images/SonyZ505.jpg</Product.photoURL>
    </Product>
    ...
  </XMI.content>
</XMI>

```

Figure 8: An XML Document Object of a Catalogue

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="start" title="Product Catalogue">
    <p align="left"><small><a href="#z505je">Sony VAIO Z505</a></small></p>
    <p align="left"><small><a href="#cp14479">IBM ThinkPad 570e</a></small></p>
    <p align="left"><small><a href="#1200x1118">COMPAQ Presario 1200</a></small></p>
    <p align="left"><small><a href="#3crwe737a">Xircom Wireless LAN PC Card</a></small></p>
  </card>
  <card id="z505je" title="SONY VAIO Z505">
    <p><small>Price: HKD 24,990.00</small></p>
    <p><small>The small size and weight of the Sony VAIO Z505 SuperSlim Pro Notebooks are quite
    deceptive. Inside this incredibly small notebook is a fully functional, powerful PC. The
    magnesium-alloy case also gives this notebook a unique look as well as incredible
    durability. The 12.1" screen gives you an extra large, full-color XGA active matrix LCD
    display.</small></p>
  </card>
  ...
</wml>

```

Figure 9: A Catalogue Deck in WML

3.3 Data Views

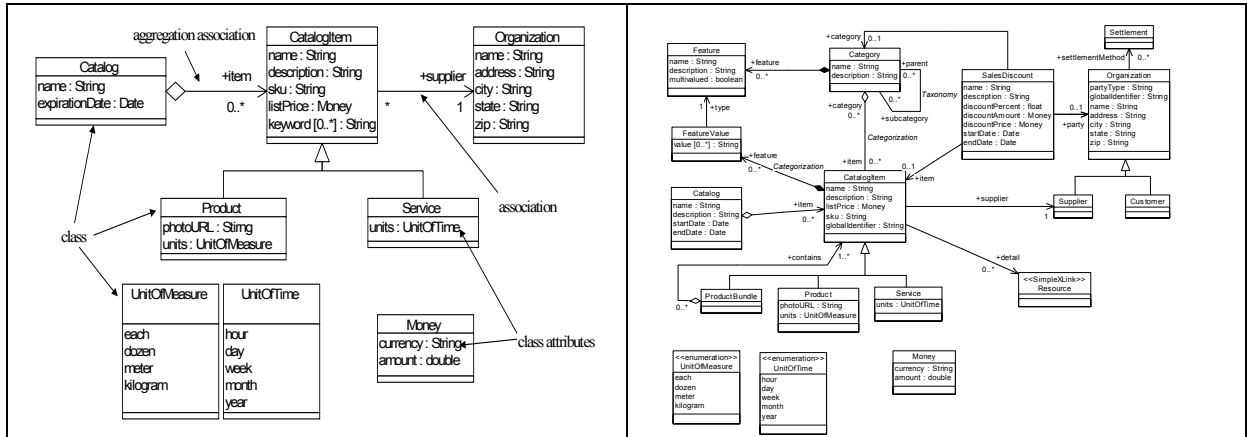


Figure 10: UML schema of a data view vs. the entire enterprise data

A data view is a set of tables comprising a projection of the enterprise data required for the enactment of some workflow views. Figure 10 presents visually the schema of a data view in UML for the *Purchase Request Approval* workflow view adapted from an example of catalogue structure in [32], together with a possible schema of the enterprise data. As shown in the figure, the data view discussed is a selection of a subset of the tables holding enterprise data as well as a projection of some table columns. Note that a data table corresponds to either a class or an association in the diagram. Table columns are represented by the class attributes. The main table in this data view is the *Catalogue* table, which consists of a number of catalogues (*CatalogItem* table) for different disciplines, such as computers equipments, office supplies and shipping companies. Each *CatalogItem* table keeps an inventory of catalogue items correspond to either a tangible product or a service. Tables *Product* and *Service* keep the details of products and services, respectively. Note that a catalogue item could fall into more than one catalogue. In other words, the relation between *Catalogue* and *CatalogItem* is a many to many relation, which is kept by a separate table *item*, which is represented as an aggregation association in the diagram. There may be more than one organization providing a catalogue item. This information is stored in the table *supplier* represented as an association. The detail of each supplier is kept in the table *Organization*. The data view represents the set of data required by the *Review Product Information* activity in the *Purchase Request Approval* workflow view. In fact, the data view is a projection and selection of the entire set of enterprise data.

4. ME-ADOME ARCHITECTURE

We extend a flexible, web-enabled workflow management system, ADOME-WFMS [7][8], into E-ADOME [5] to strengthen the external interface layer to interact with different types of agents and users over the Internet more effectively. A recently update is the employment of Web services [9] support to replace a traditional web-server. We further extend E-ADOME to ME-ADOME to support mobile users, and in particular, more comprehensive workflow views. As shown in Figure 11, the ME-ADOME environment can be divided into the following layers. The *ADOME / OODBMS Layer* was developed by integrating a CLIPS rule engine [10] and an ITASCA OODBMS [19] to allow for more adequately deal with data and knowledge management requirements of advanced information management applications, especially WFMSs. This layer is the backend database and provides *data view* support to the WFMS. The *ADOME-WFMS Layer* is a flexible WFMS built upon ADOME facilities, supporting effective management of agents, on-line workflow evolution, and automatic and cooperative exception handling [7]. This layer provides *workflow view* support to the WFMS. The *Internet Interface Layer* is the enhancement layer to the WFMS for ADOME-WFMS to interact with

various types of external agents and users through the Internet. This layer provides *user interface view* support to the WFMS for *External Clients and Agents*, which provide services or solving problems.

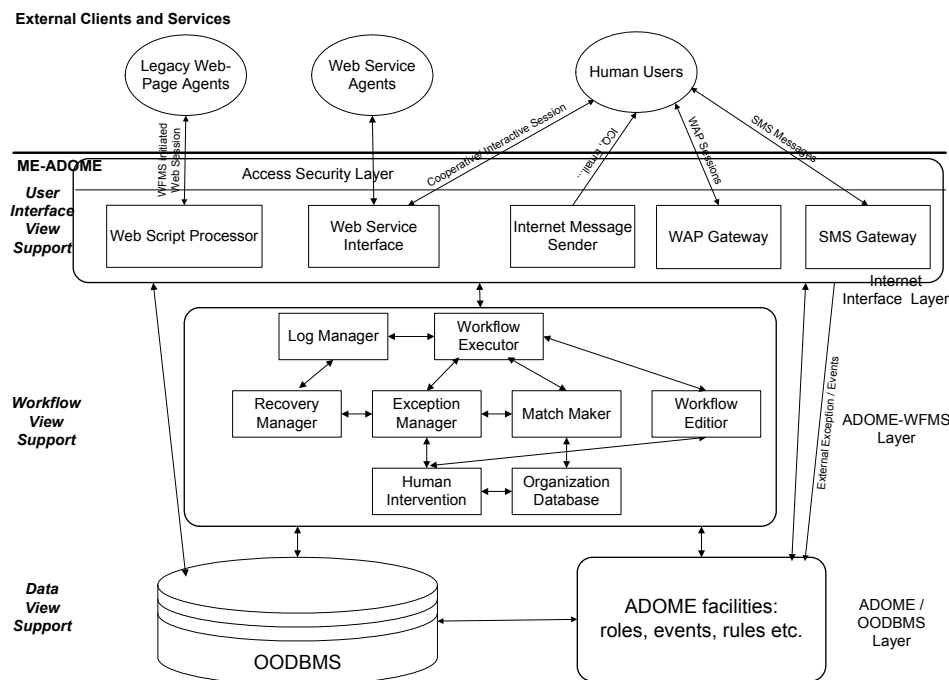


Figure 11: ME-ADOME architecture

The ME-ADOME *Agent Interface Layer* allows the ADOME-WFMS to interact with agents and users through the Internet. Since these users and agents likely require different interfaces, the *Internet Interface Layer* is designed to be highly adaptable. Furthermore, an *Access Security Sub-Layer* is also added to handle security issues of external communications. The *Internet Interface Layer* has the following three components (cf. Figure 11). The *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. Note that these users and agents may be internal or external to the organization.

Web Service Interface module enables ME-ADOME to communicate with other WFMSs to allow for more advanced task execution and control in foreign WFMSs. At the same time, it also supports users to interact with ME-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. *Web Script Processor* enables ME-ADOME to initiate an automatic conversation script with other interactive, web-based service providers without compatible software API, including most online ordering web pages or service report forms. Without this facility, the WFMS need a staff to perform this activity manually *WAP Gateway* enables ME-ADOME to carry out interactive sessions with mobile users on handsets and networks with WAP support. *SMS Gateway* enables ME-ADOME to send alert message to mobile users on handsets and networks with SMS support. In addition, the *SMS Gateway* can receive SMS message from these users too. Hence, SMS conversions can simulate interactive sessions with limited length and complexity.

We separate user alerts from user sessions with the WFMS to improve the flexibility. Online users will be alerted by ICQ, with the task summary and reply URL as the message content. If the user is not online or does not reply within a pre-defined period, the WFMS will send the alert by email. At the same time, another alert may be sent via SMS to the user's mobile phone. Whatever the alert channel has been, the user need not connect to WFMS on the same device, or even on the same platform. For example, after receiving a SMS alert, the user may use his handset to connect to the

WFMS via WAP, or he/she may reply with an SMS message. Alternatively, the user may find a PC with Internet connection or use his / her PDA to connect to the WFMS.

5. DESIGNING VIEWS IN THREE TIERS

Based on our experience from the purchase request workflow, we present a methodology for designing views in three tiers in this section. The scenario of extending an existing web-based application to support mobile users is usually more typical than designing a multi-platform application from scratch. In addition, the following discussion is also applicable to phased projects in which usually standard web-based application are developed as the first phase, and supporting mobile users as subsequent phases. We propose the design of the views in three tiers to be carried out in the following major steps:

1. Design workflow (process) views based on application, user and platform requirements
2. Design data views based on the requirement of the workflow views
3. Design user interface views based on platform dependent restrictions

5.1 Designing Workflow Views

Like software design, it is unlikely that there is a standard recipe for mechanical design of workflow views that is universally applicable to all business processes. In order to capture the requirements for mobile users, one can adopt a use case driven approach derived from object-oriented software design [21]. A use case refers to a non-trivial task of added values to an end user in a workflow process. A workflow view could then correspond to one or multiple related use cases obtained through object-oriented analysis. For example, we can concentrate on the difference of the requirements for the mobile users from those under standard browsers. The analysis will lead to different use cases in an object-oriented software design, which in turn results in the formulation of different workflow views. In general, a complete detailed business process is usually too complicated for a mobile environment. Therefore, typical requirements are simplification of the process, reordering of work steps, delegation of tasks (work steps) to other personal, etc. For example, the process “Budget Check” may be delayed, delegated to another staff with a standard browser, or carried out by a computer program.

Mobile applications may be enhanced by using the awareness of the user’s location, and location dependent information available and relevant to the application, e.g., mobile workforce management systems and provision of information to mobile users.

Exception handling due to unavailability of the required personnel, network delay, interrupts and disconnections, etc., should also be considered. In the worst case of aborting a workflow, we should consider additional requirements for compensation workflows (e.g., an order cancellation) under a mobile environment.

These differences are compared with the standard workflow to formulate views. We should identify similar or identical tasks to maximize reuse, and in particular, consider the possibility of customizing them with data views and user interface views instead of rewriting them. For example, we have reused the tasks “Review PR Form” and “Review Product Information” as described in the previous sections. Designing Data Views

5.2 Designing Data Views

With a workflow view defined, we can proceed to analyze its data requirement. Each work step requires data from the database in some form. In particular, we should identify mandatory fields,

optional fields and fields that are to be skipped in the view, in order to cope with the simplification required for mobile users. However, additional fields that have to be computed for summarizing information and knowledge may be required. In case that the mobile user cannot provide mandatory information or input them effectively, we may need to modify the workflow so that the mandatory information can be provided later or by someone else. Therefore, like designing software, the process is not a linear one. In addition, security requirements should also be considered, e.g., sensitive information may have to be restricted to users within the office or to those of pre-approved locations.

5.3 Designing User Interface Views

With the data requirement of individual work steps determined, we can proceed to design the user interface views that are customized for different computing platforms. Section 2.2 has already discussed some details of the limitations of mobile platforms. In summary, we usually need to remove graphics or reduce the resolution, provide panning and zooming, shorten fields or provide summarized ones instead, break one web page into several screens, etc. For user input, we should consider the difficulties in entering data (especially typing) on mobile devices, and provide menu selections as far as possible.

5.4 Discussion

As a basis for reference and comparison, we consider the user case of the system via a standard web browser. For PDA interface, the main problem is just a smaller screen, some of which may be black-and-white. If the original full-function user interface is too complicated (e.g., too many unnecessary features or high resolution screen layout), another simplified *user interface view* is probably required. Pictures and documents may require to be shown in lower resolution and documents may be outlined and level-structured. Panning and zooming (supported by most browsers) should also help. In addition, *data views* may also be employed to hide less important data field or to show alternate summary columns. Though the simplified views are less informative, the same workflow is still most probably fine.

For users on a WAP interface connecting to the WFMS via a WAP gateway, the screen is extremely small. A *user interface view* is mandatory to map the original one into WML. Data views are also required to provide only the critical information and to summarize it. Furthermore, a workflow view providing a simplified version of a process is often necessary since complex activities are not at all suitable to be carried out in such an environment. For instance, a mobile user may wish to skip some alternative choices or parts of a document.

For users on a mobile phone but cannot be connected via WAP, simulated sessions can be carried out through a SMS gateway by exchanging SMS messages. A totally different set of views is required, viz., a *workflow view* specifying the SMS dialogue sequence and options, a *data view* providing highly summarized data in simple messages, and a *user interface view* for the SMS technology. Usually, the user can only make very simple decisions or feedback. As this kind of processes are often error prone and may have security problems, it is often advisable disallow critical options in these workflow views.

In general, it is often difficult to tell if a user can tolerate a complex workflow because of the operating environment or even due to the user's mood, which cannot be determined merely with observation facts, such as the mobile platform and physical location. Therefore, it is always a good idea to allow users to choose their desired view options whenever feasible.

6. RELATED WORK

Mobile workflow is now only at a starting stage. The WHAM (workflow enhancement for mobility) prototype [22] supports mobile workforce and applications in a workflow environment, with a focus on network connectivity and the mobility of workflow resources. Tjoa [30] introduced an Java Border Service Architecture, which is an abstract layer between presentation and application logic of an application, to handle mainly user interface issues of mobile devices, using workflow as an example. As for commercial products, Staffware [28] has recently introduced, WAP Business Process Server. However, all of them do not support platform specific workflow adaptation or integrated platform independent solution. Neither do they support view mechanisms.

Our ideas in workflow views for cross-organizational business interaction and e-contract enactment have been presented in [5]. This approach has been motivated by views in object-oriented data models, which can be dated back to [11], and in particular by imaginary objects in [2]. Gardarin [15] discussed federated OODBMS and views for objects in a distributed environment. Liu and Shen [25] presented an algorithm for workflow view construction and verification, but does not discuss any of its applications. Van der Aalst and Weske [1] introduced the concept of inheritance of a public workflow from a private workflow to achieve interoperability in a cross-organizational e-commerce environment.

Dartflow [3] is one of the first web-based WFMS, using transportable agents, CGI and Java technologies. Eflow [4] is one of the closest commercial systems with features like ME-ADOME in handling e-Services. However, Eflow does not address matching of agents directly with activities. Instead, it uses the concept of generic service node and service selection rules. Currently, several commercial WFMSs such as TIB/InConcert [29] and Staffware 2000 [27], provide web user interface too. In addition, I-Flow [13] has a Java workflow engine. WW-flow [23] 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 [4] and Crossflow [16].

However, not all the above-mentioned WFMSs support various kinds of interactions with different kinds of users and agents, as in the ME-ADOME Internet interface layer. In particular, we have developed a novel three-tier view-based approach to supporting mobile workflows. Compared with the systems close to us, ME-ADOME has the most features and can support E-services, E-contracts, and mobile users and agents on the Internet.

7. CONCLUSIONS AND FURTHER WORK

This article has presented an advanced workflow environment with pragmatic features in supporting various types of users and agents, in particular mobile ones, over the Internet. We have illustrated, in the context of ME-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, ME-ADOME provides an improved environment, which can adapt to changing requirements of mobile users, with extensive support for reuse. The main contribution of this article has introduced a novel three-tier view-based approach (*viz.*, *workflow view*, *user interface view* and *data view*) to provide flexible support for mobile users on different platforms. Further note that, ME-ADOME specification of workflow views is based on standardized Workflow Management Coalition workflows, many of the techniques presented in this article can thus be applicable to any WFMS. In addition, the concepts of *user interface view* and *data view* should be widely applicable to a wide range of mobile applications.

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. We consider further research issues on interfacing and interoperability important for extending the applicability of an advanced WFMS engine, which include: expanding the possible interfaces and coordinating different types of agents, graphical workflow evolution tools, and inter-operating with other WFMSs. In addition, we are interested in the application of ME-ADOME in various advanced real-life e-commerce environments, such as procurement, finance, stock trading and insurance. In particular, we are looking into location depend application, such as mobile workforce management and mobile customer relationship management (CRM) applications. We are developing a more unified way to exchange information, including workflow views, with other agents, through XML. We are also interested in wrappers to interface with legacy software agents. ME-ADOME is currently being built on top of the ADOME-WFMS prototype system, with a various user interfaces to accommodate the whole range of activities on different platforms.

REFERENCES

- [1] 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.
- [2] S. Abiteboul and A. Bonner. Objects and Views. In Proceedings of ACM SIGMOD Conference, pp238-247, 1991.
- [3] 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.
- [4] F. Casati, et al. Adaptive and Dynamic Service Composition in eFlow. HP Laboratories Technical Report HPL-2000-39, March 2000.
- [5] D.K.W. Chiu, S.C. Cheung and E. Kafeza. View-based Support for Mobile Workflow. First International Conference on Mobile Business, Athens, Greece, July 2002.
- [6] D.K.W. Chiu, K. Karlapalem, Q. Li and E. Kafeza. Workflow Views Based E-Contracts in a Cross-Organization E-Service Environment, Distributed and Parallel Databases, Kluwer Academic Publishers, 12(2-3):193-216, 2002.
- [7] D.K.W. Chiu, Q. Li and K. Karlapalem. A Meta Modeling Approach for Workflow Management System Supporting Exception Handling. *Information Systems*, Pergamon Press, Elsevier Science, 24(2):159-184, 1999.
- [8] 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, 2001.
- [9] V. Chopra, et. al. Professional XML Web Services, Wrox Press, 2001.
- [10] GHG Corp. Clips Architecture Manual, Version 5.1. Jan 1992. (available at <http://www.ghg.net/clips/CLIPS.html>)
- [11] U. Dayal. Queries and Views in an Object-Oriented Data Model. In Proceedings of 2nd International Workshop on Database Programming Languages, pp 80-102, 1989.
- [12] R. A. Elmasri and S. B. Navathe. Fundamentals of Database Systems, Addison-Wesley, 3rd edition, 2000.
- [13] Enix Consulting Limited. An Independent Evaluation of i-Flow Version 3.5, 2000 (available at <http://www.i-flow.com>).
- [14] H. Evans and P. Ashworth. Getting Started with WAP and WML, Sybex, 2001.
- [15] 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.
- [16] 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.
- [17] A. ter Hofstede, M. Orłowska and J. Rajapakse. Verification Problems in Conceptual Workflow Specifications. *Data & Knowledge Engineering*, Pergamon Press, Elsevier Science, 24(3) (1998) 239-256
- [18] Hewlett Packard. Changengine Admin Edition (AdminFlow) Process Design Guide, 1998.
- [19] Itasca Reference Manual, Ibx Corporation, 1994.
- [20] ICQ. <http://www.icq.com>
- [21] I. Jacobson, G. Booch and J. Rumbaugh. The Unified Software Development Process, Addison Wesley, 1999.
- [22] J. Jing, K. Huff, B. Hurwitz, H. Sinha, B. Robinson and M. Feblowitz. WHAM: Supporting Mobile Workforce and Applications in Workflow Environments. 10th International Workshop on Research Issues in Data Engineering, San Diego, California, IEEE Press, pp 31-38, 2000.
- [23] 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.
- [24] Y.-B. Lin and I. Chlamtac. Wireless and Mobile Network Architectures, John Wiley & Sons, 2000.
- [25] 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

- [26] D. McCarthy and S. Sarin. Workflow and Transactions in InConcert. IEEE Data Engineering,16(2) (1993) 53-56
- [27] Object Management Group. Foreword UML specification 1.4, September 2001.
- [28] Staffware Corporation. Staffware Global - Staffware's Opportunity to Dominate Intranet based Workflow Automation, 2000, <http://www.staffware.com>
- [29] TIBCO Software Inc., which has acquired InConcert Inc., <http://www.tibco.com>
- [30] A. M. Tjoa, R. R. Wagner, A. Al-Zobaidie. On Integrating Mobile Devices Into A Workflow Management Scenario. Proc. 11th International Workshop on Database and Expert Systems Applications (DEXA 2000), IEEE Press, pp 186-192, 2000.
- [31] Workflow Management Coalition. The Workflow Reference Model. (WFMC-TC-1003, 19-Jan-95, 1.1)
- [32] D. Carlson, Modeling XML Applications with UML, Addison-Wesley, 2001.
- [33] <http://www.w3.org/XML/>