"Designing Workflow User Interfaces with UsiXML"

Guerrero Garcia, Josefina ; Vanderdonckt, Jean

Designing Workflow User Interfaces with UsiXML

Josefina Guerrero Garcia, Jean Vanderdonckt
Université catholique de Louvain, Louvain School of Management, Place des Doyens, 1
B-1348 Louvain-la-Neuve (Belgium)
{josefina.guerrero, jean.vanderdonckt}@uclouvain.be

ABSTRACT

Supporting business processes with the help of workflow systems is a necessary prerequisite for many companies to stay competitive. An important task is the specification of workflow, i.e. these parts of a business process that can be supported by a computer system. This paper is about the definition and development of user interfaces for workflow information systems. XML-based user interface description languages express various aspects of the user interfaces, including the abstract and concrete elements of the user interface, the tasks to be performed by the users, and the user interface dialogue. We have developed a framework for expressing workflow aspects, and using UsiXML for rendering the user interfaces.

Categories and Subject Descriptors

D2.2 [**Software Engineering**]: Design Tools and Techniques – *Modules and interfaces; user interfaces.* D2.m [**Software Engineering**]: Miscellaneous – *Rapid Prototyping; reusable software.* H.1.2 [**Information Systems**]: Models and Principles – *User/Machine Systems.* H5.2 [**Information interfaces and presentation**]: User Interfaces – *Prototyping; user-centered design; user interface management systems (UIMS).*

General Terms

Design, Experimentation, Human Factors, Verification.

Author Keywords

Workflow, user interface description language, task model, organizational model.

INTRODUCTION

In every company there are procedures aimed at providing increased efficiency, consistency, and quality. Among them are concepts such as business process and workflow management. Their objective is to clarify, improve, and coordinate complex activities and interactions. The goal of a business process is to deliver specific products or services while ensuring the organization's overall interests. Examples for business processes are processing purchase orders over the phone, processing insurance claims, and processing a car registration. The objective of *workflow management* is to support the execution of business processes. The parts of a business process that can be supported by a computer system are called *workflows*. A workflow management system (WFMS) is a computer system that implements workflow management functionality.

In this context, an important aspect to consider is the design of the user interfaces of the system. In the area of workflow there has been increasing interest in developing methods and languages to design user interfaces, as the work of [1] and [12]. In Human-Computer Interaction, a user interface description language express various aspects of the user interfaces, including the abstract and concrete elements of the user interface, the tasks to be performed by the users and the user interface dialogue. Nevertheless, there are several opportunities to improve on these languages; in this paper, we will explore a systematic way to design the user interfaces for a workflow information system.

The workflow model defines what processes and tasks need to be fulfilled and their possible ordering, hence the workflow model is a "framework" for creating task models; hence the task model is suitable for designing user interfaces.

The remainder of this paper is structured as follows: Section 2 presents an overview of the related work. Section 3 introduces a model for specifying the workflow, and then we describe a method for designing workflow user interfaces based on UsiXML (Section 4). Section 5 presents a case study and the software supporting the method. Section 6 summarizes our work deriving conclusions and future work.

RELATED WORK

In order to provide a meta-model of workflow concepts which are related to user interface issues, this section presents an overview of the existing literature. It has been divided in three related parts: the workflow characteristics, the task model concepts, and the user interface description languages to generate user interfaces. There is a plethora of methods, tools, and models in the literature, including open source systems and vendors systems, research into all of them is not the objective of this section, and we just present some of them.

Workflow theory

Workflow models focus on how work is done to accomplish organizational goals; it defines how task, information, and documents are passed from one participant to another in the organization [28]. According to [16], the essential workflow characteristics are: *tasks/activities* that are performed by (role-playing) *persons,* using supporting *tools* that give access to a variety of *shared information resources*. Due to the importance of workflow nowadays, several workflow notation, descriptions, and software support have been proposed to design and specify it.

In Faure, D., Vanderdonckt, J., (Eds.), *Proc. of 1st Int. Workshop on User Interface Extensible Markup Language UsiXML'2010* (Berlin, 20 June 2010), Thales Research and Technology France, Paris, 2010. ISBN 978-2-9536757-0-2

Some notations used are: *Petri Nets* [22]as a technique for modeling and analyzing processes; *Statechart Diagrams* [27] is a graph that represents a state machine describing the response, of an object of a certain class, to the receipt of outside stimuli; *Business Process Modeling Notation* (BPMN) [18] is a standardized graphical notation for drawing business process in a workflow; and *UML Activity Diagrams* [4] provides a view of the behavior of a system by describing the sequence of actions in a process. UML activity diagrams are intended to model both computational and organizational processes. Currently, several models and design methods support the development of complex workflow-based applications providing notations for business process and including tool support. Some of them are: *The Progression Model* [21] has incorporated some of the managing concepts of workflow to increase the flexibility in information systems; *Microsoft Windows Workflow Foundation* (WWF) [5] is an extensible framework for developing workflow solutions on the Windows platform. It provides a single, unified model to create end-to-end solutions that span categories of applications, including human workflow and system workflow; *WebSphere® MQ Workflow* (IBM) [10] supports long-running business process workflow as they interact with systems and people. Automates and tracks business process in accordance with business design; *Workflow on Intelligent and Distributed database Environment* (*WIDE)* [3] defines an advanced conceptual model for describing both the flow of activities and the organizational environment in which these activities are performed; *Business Process Visual ARCHITECT* (BP-VA) [26] is a visual modeling tool that provides the most extensive support for BPMN.

Due to the large amount of existing workflow products we came to a point where it is very difficult to analyze and compare their capabilities on a common scheme. However, they can be gathered in a collection of workflow patterns that provide the basis for an in-depth comparison of commercially available workflow systems. *Control-flow patterns* [23] identified useful basic routing constructs such as sequence, parallel split, synchronization, exclusive choice. From a data perspective, there is a series of characteristics that occur repeatedly in different workflow modeling paradigms. *Workflow data patterns* [19] are aimed at capturing the various ways in which data is represented and used in workflows. *Workflow resource patterns* [20] correspond to the manner in which tasks are allocated to resources, that is any entity that is capable of achieving some work unit.

Task theory

A common definition for a task is "an activity performed to reach a certain goal" [25]; task models play an important role because they indicate the logical activities that an application should support to reach user' goals. While the purpose of task analysis is to understand what tasks should be supported and what are their related attributes, the aim of task modeling is to identify more precisely the relationships among such tasks. Task models are explicit representations of user tasks that can help support certain rigorous forms of task analysis. In [6] we discuss some well-known and widely used task notations, examining which characteristics they exhibit and which attributes they cover. Task models play an important role in UI design because they support the systematic representation of the user activity as opposed to the system activity.

User interface description languages theory

Model-based user interface design is intended to assist in designing UIs with a more formal computer supported methodology; a user interface description language (UIDL) is intended to capture the details of what a user interface could or should consist. There are solutions for developing UIs that are based in eXtensible Mark-up Language (XML). There is a plethora of user interface description languages that are widely used, with different goals and different strengths. On one hand we have software vendors UIDLs and, on the other hand, there are free license UIDLs to use. A review of user interface description languages was produced [9] that compares a significant selection of various languages addressing different goals, such as multiplatform user interfaces, device-independence, and content delivery. UsiXML [12] is a XML-compliant markup language capturing the essence of what a UI is or should be independently of physical characteristics. It has been selected as the UIDL to be used in the remainder of this work because of its capabilities of extensiveness, availability, central storage of models, and its model-driven approach.

A WORKFLOW MODEL

After reviewing the available literature, we can identify the elements needed to model the workflow. We propose a model called *FlowiXML* (user interfaces to work**flow** based on Us**iXML**) [7] to support workflow information systems, which are advocated to automate processes based on requirements and processes of the organization. The methodology applies to:

1) Integrate human and machines based activities, in particular those involving interaction with IT applications and tools.
2) Identify how tasks are structured, who perform them, what their relative order is, how they are offered or assigned, and how tasks are being tracked. Figure 1 presents an overview of the meta-model. FlowiXML is divided in different models.

Workflow model

The workflow model describes how the work in organization flows by defining models of: process (what to do?), tasks (how to do it?), and the organizational structure (where and who will perform it?). Also, it contains information relevant to controlling and coordination of the execution of its constituent tasks (e.g. required skills, possible actors, and execution requirements). A workflow model has at least one process and each process has at least two tasks. Workflows are described with a *name* and *ID*.

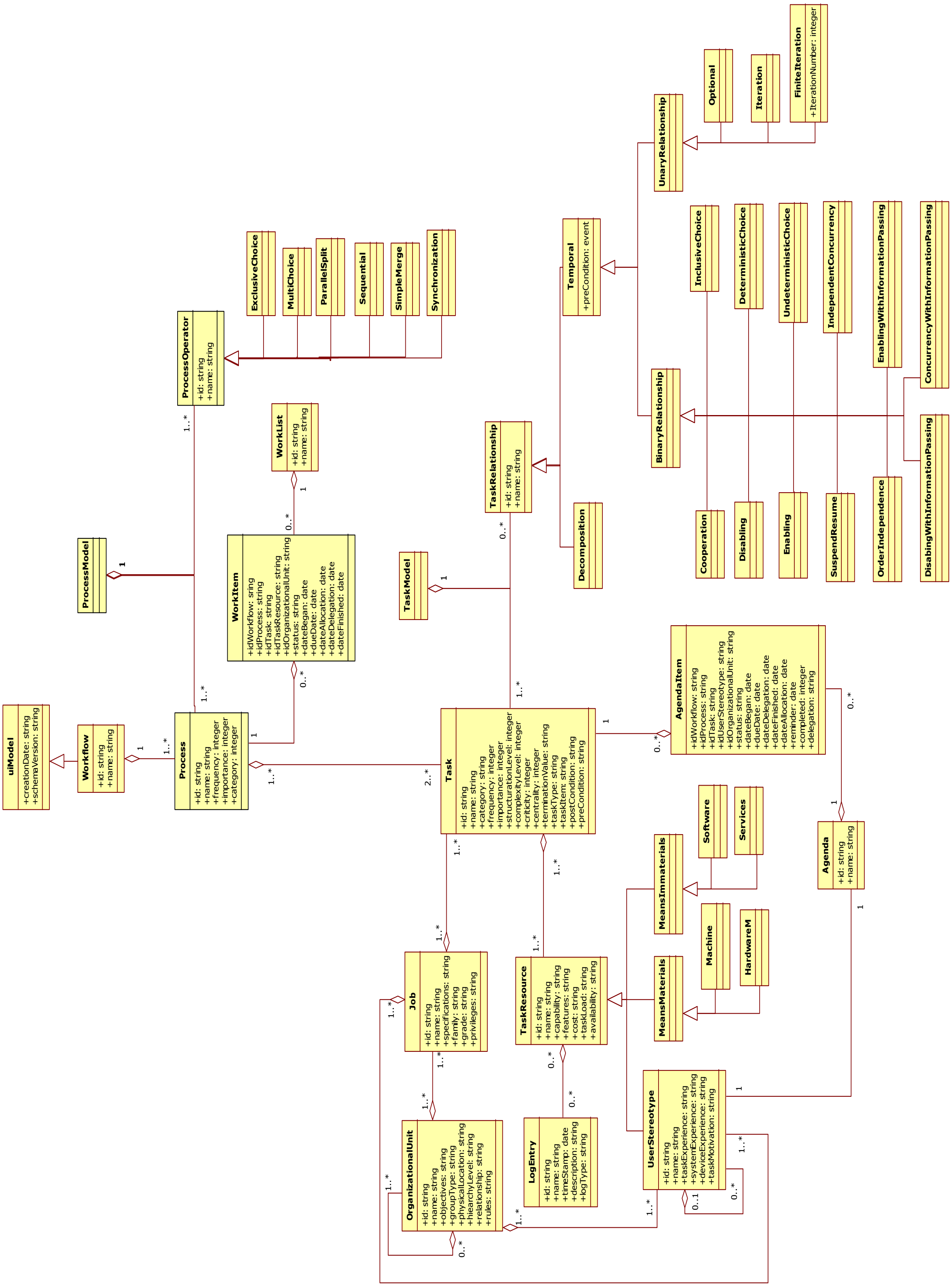


Figure 1. Workflow meta-model.

Process model

The definition of a process indicates the ordering of tasks in time, space, and resources. The representation of this model is an adaptation of the Petri Nets notation proposed in [22]and is compatible with the workflow resource patterns proposed in [20]. The process model is composed of:

- *Process*. A process consists of a number of tasks and a set of relationships among them. The definition of a process indicate which tasks must be performed and in what order.
- *Process operators*. They indicate the different ways in which the processes could be executed.
- *Work list*. It is a list of work items related with a given workflow.
- *Work item*. It is the representation of the work to be processed.

Task model

Task models are used to collect the requirements of a workflow system. A task is an activity that has to be performed by users to reach a given goal related to the processes. Introducing task models description to the workflow models corresponds, but is not limited, to the following reasons:

- Task models describe, opposed to process models, end users' view of interactive tasks while interacting with the system. This allows describing how a task is performed.
- It is true that in a process model we can add the detail desired, with process hierarchies, to represent a detailed task description. However, we consider that specific temporal operators, iteration, suspend/ resume, applied to task, can be more naturally defined in a task model rather into a process model, that implies the creation of dummy transitions.

The task model described in [12] [23] benefit from a conceptual contribution; the benefits consist mainly in added some task attributes and relationships. The condition is an attribute not considered in previous version and essential to the occurrence of the task.

- *preCondition*. Before a task can be performed it must fulfill certain conditions.
- *postCondition*. A task can be considered finished if it fulfills the necessary requirements.

The binary relationships that were introduced are:

- *Cooperation*. These relationships specify the cooper tion that could be between two or more tasks.
- *Inclusive choice*. These relationships specify two tasks that: both could be executed or just one of them or neither of them.
- *Disabling with information passing*. These relationships occur if one task is completely interrupted by another task; and the information produced in the first task is used as an input for the second task.

Organizational components

Although information systems are an integral part of organization; the key elements of an organization are its people, structure, business processes, politics, and culture. We propose an organization framework representing the places where work is performed and the users that perform it. This part contributes to UI adaptation to different categories of users and security of IS by blocking access to UIs when the user does not have the permission to perform the task.

The attributes of this model are:

- *Organizational unit*. An organizational unit is a formal group of people working together with one or more shared goals or objectives. It could be composed of other organizational units.
- *Task resource*. A task resource is an entity that is directly or indirectly involved in carrying out the work.
- *User stereotype*. This class represents the set of users sharing the same values. Each user stereotype may in turn be decomposed into sub-user stereotypes.
- *Means materials*. It is a type or resource that is physically tangible and is a non human resource.
- *Means immaterial*s. It is a type of resource that is physically intangible; it does not have a material form or substance.
- *Log entry*. LogEntry describes specific characteristics that resources may possess. Each resource may have a log Entry associated with him.
- *Job*. Jobs are the total collection of tasks, duties, and responsibilities assigned to one or more positions which require work of the same nature and level.
- *Agenda item*. Agenda items are the tasks that a userStereotype has to perform.
- *Agenda.* The agenda is a list of agendaItem that are assigned to userStereotypes. A userStereotype has one and only one agenda and an agenda belongs to one and only one userStereotype.

Mapping model

UsiXML is composed of various models [23], among them the mapping model serves to gather a set of pre-defined, inter-model relationships that are semantically related.

This mapping model benefits from a conceptual contribution in order to link the tasks with resources (Figure 2). Based in [20] we use the workflow resource patterns to capture the various ways in which tasks are allocated. We extend the mapping model by adding:

- *Is grafted on*. It grafts a task on another one. This relationships is useful when a task (*Tj*) has been executed, and a task complementary (*Ti*) is defined to realize the first task. *Ti* is completely autonomous to *Tj.*
- *Is defined by*. It refers to a task defined by a userStereotype.
- *Is allocated to*. It corresponds to a task that is assigned to a resource.

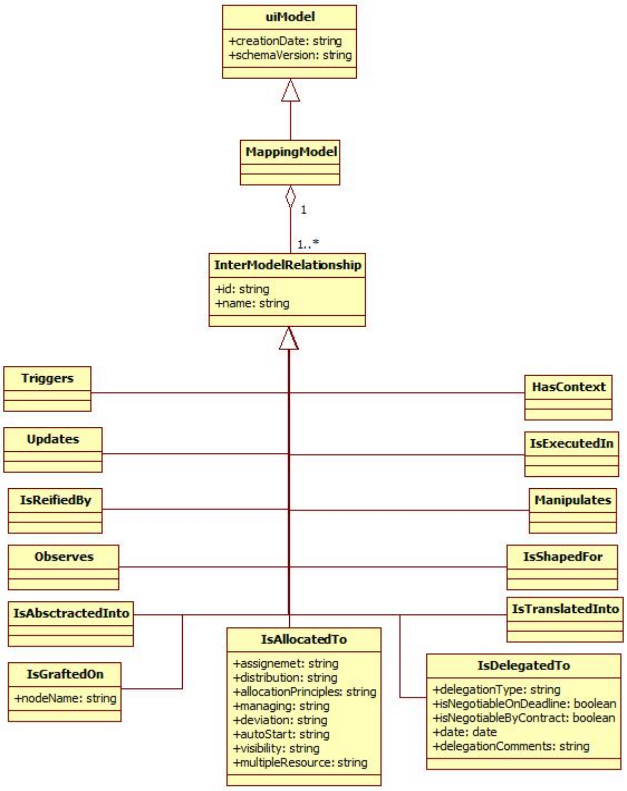


Figure 2. Mapping model.

Workflow users

When a workflow is designed, several *userStereotypes* are involved in the process:

- The *Workflow designer* is in charge of the conceptual process and the drawing of the organization. Direct communication is needed to the *workflow manager* of the organizations to have the global view of the processes to be modeled. The results provided are mock-up of the workflow of the processes to be modeled.
- The *Workflow manager* is a person who is responsible for the handling of a whole workflow. Through the modeling the knowledge required to model the organization is captured with meetings with supervisors, workers, etc.
- The *Process manager* is the person who is responsible for a particular process.
- The *end user* is any *userStereotype* that belongs to the organizational model (internal) or is part of other organization (external). It can be the manager, an economist, the lawyer, any valid user declared in the workflow specification.

The above roles could be assigned to different persons or combined into one single person. Apart from the conception of a workflow system, the final use of the system could impact several levels in the hierarchy of the organization.

DEVELOPING WORKFLOW USER INTERFACES

Developing user interfaces from a workflow specification has several dimensions to be tackled. We need user interfaces to:

- Support user's tasks specified in task models
- Support user's communication with agendas that must be updated accordingly as users received a new item or when they finish a task
- Support tasks allocation, resource patterns, by different means such as: allocation, offering, delegation.

In order to develop UIs, we consider (Figure 3):

- Elicitation of the organization. We assume that there are means to collect information that will serve as input.
- Workflow modeling of the problem which includes the definition of: the organizational units, jobs and user stereotypes, process models, workflow allocation patterns and task models.
- Mapping the workflow specification into a workflow information system.
- Manage the workflow through the workflow manager.

.

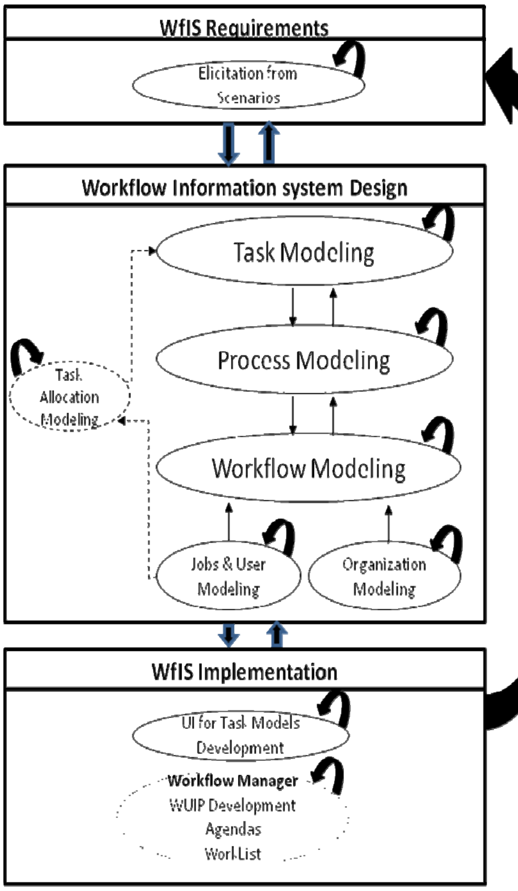


Figure 3. Method overview.

Forward and backward arrows denote the propagation of information from one model to another. For instance, a new task model must make available a task for a process model and vice versa a new task in a process model might be detailed with a task model. Jobs, user stereotypes and organizational modeling just affect the workflow model. Then the workflow model makes them available for process modeling and task modeling. This particular aspect of concepts propagation was significantly useful for the software tools that support our method. We will explain the steps of this method, even they are presented consecutively, it is not mandatory to follow this order.

Step 1: *What to do?* **Processes specification**

As we pointed before, the definition of a process indicates which tasks must be performed and in what order. As notation we use Petri Nets, a tool for modeling and analyzing processes.

Step 2: *How to do it?* **Tasks specification**

For each process a task model can be specified to describe in detail how the process/task is performed. By exploiting task model descriptions different scenarios could be conducted. Each scenario represents a particular sequence of actions that can successfully be performed to reach a goal. Task models do not impose any particular implementation so that user tasks can be better analyzed without implementation constraints.

Step 3: *Where to do it?* **Organizational units' specification**

We introduced an organizational unit concept; it describes the places where work is carrying out. It can be a physical chamber in a builder (e.g. an operating room), but it can also be a complete floor in a hospital. The elements corresponding to the actual organizational structure in a specific organization are specified during the organization design.

This part contributes to UI adaptation to different categories of users and security of IS by blocking access to UIs when the user does not have the permission to perform the task.

Step 4: *Who will carry out it?* **Job and user specification**

This step consists in the description of all the users involved in the performance of tasks. Jobs are ways to structure the crew of people inside the organization. It involves the complete collection of knowledge and practices needed by a definite human resource to perform a task.

Step 5: *Whom?* **Applying workflow resource patterns**

Actual assignment of tasks to resources is performed according to workflow resource patterns [20]. The way in which tasks are advertised to resources is essential; therefore we need to consider the different states that task goes through, from creation to termination. We propose a life cycle (Figure 4) of the task from the time that it is created to final completion (or failure or cancel).

Workflow resource patterns have been identified that capture the different manners in which resources are presented and used in workflows. The rationale for identifying these patterns was the need to master the many ways according which work can be distributed.

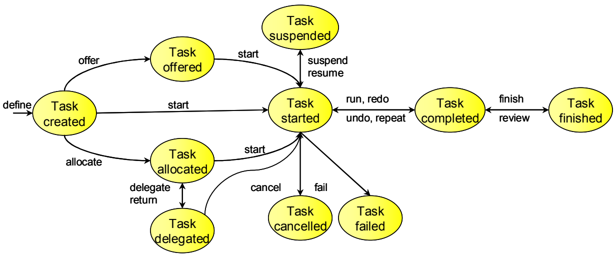


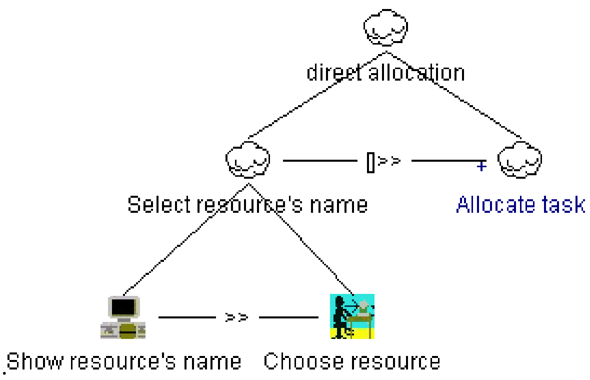
Figure 4. Task life cycle.

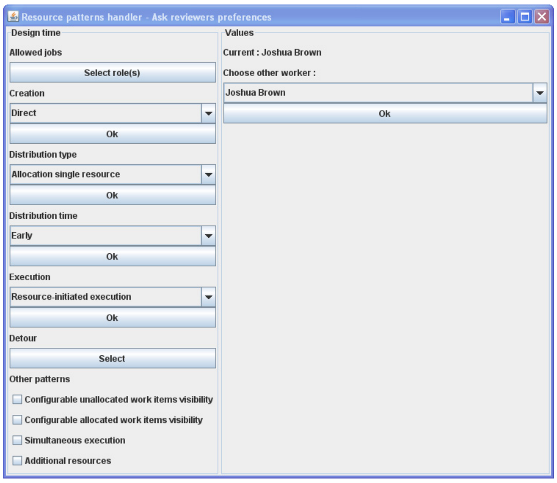
In [8] a collection of user interface design patterns for workflow information systems is presented. We give below only a snapshot (Figure 5) for facilitating the understanding and for illustration purpose.

Step 6: Mapping the workflow to user interfaces

Once the workflow has been defined, we part from the task models for developing the workflow UIs and then we follow the different steps until provided the final user interface (Figure 6). We completely rely on UsiXML which is compliant with the Cameleon Reference Framework [1].

Name: Direct allocation
Identifier: R-DA
Synopsis: The ability to specify at design time the identity of the resource that will execute a task
Strengths: To prevent the problem of non-suitable allocation
Weakness: No opportunity to change the resource if he is not available to perform the task
Opportunities: To ensure task is routed to specific resource
Problem: This pattern effectively defines a static binding of tasks to a single resource
Solution: Probably the use of deadline and escalation mechanisms when the resource becomes overload and cannot deal with his assigned workload in a reasonable timeframe
Example: "Ask reviewers preferences" task must only be undertaken by "Joshua Brown"





Resource patterns handler - Ask reviewers preferences

Design time

Allowed jobs

Select role(s)

Creation

Direct

Ok

Distribution type

Allocation single resource

Ok

Distribution time

Early

Ok

Execution

Resource-initiated execution

Ok

Detour

Select

Other patterns

□ Configurable unallocated work items visibility

□ Configurable allocated work items visibility

□ Simultaneous execution

□ Additional resources

Values

Current : Joshua Brown

Choose other worker :

Joshua Brown

Ok

Figure 5. Direct allocation pattern.

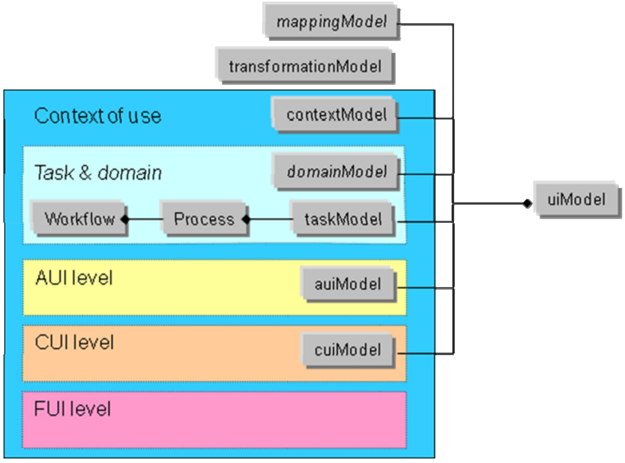


Figure 6. Cameleon Reference Framework and UsiXML.

Case study and workflow editor

In order to support the generation of UIs from a workflow model, a workflow editor has been developed. The case study will be elaborated and described among to the use of the editor and the concepts involved in each phase.

Identification of workflow, process and task. The method starts from a scenario describing the case study to be addressed in a textual format containing the existing or envisioned system with as much details as possible depending on the information available. The way this information is collected and its completeness in not in the scope of our research. Modeling workflow, processes and tasks consists of a fundamental activity that initiates user-centered design in user interface development. It is therefore important to reach the best possible model and that modeling activities remains consistent when the modeler changes. For this purpose, we introduced a set of criteria; see Table 1, in order to identify models during modeling in an unambiguous way that result into a model exhibiting desired properties of quality such as completeness, consistency. In addition, starting and stopping criteria provide designers with guidance on when and how to start and finish each model.

What? *Workflow specification*. The workflow specification, depicted in the process model, takes place inside of the organizational unit framework. Concretely, the workflow represents the business process and determines the right resource for the right task at the right time (Figure 7).

Table 1. Identification criteria.

	Time	Space (location)	Resource	Type
Workflow	Series of time periods	Different locations; same organization	Same or different groups of resources	-
Process	Series of time periods	Different locations	Within groups, group as a whole, or among groups	Primary (production), secondary (support), or tertiary (managerial)
Task	Same time period	Same location	One or two types of resources	User, interactive, system, abstract, or machine task

Where? *Organizational units' specification*. This step consists in specifying the location in which the work must be done. Organizational units' attributes are then specified in the editor and graphically the workflow designer identifies the different components of the organization. Organizational units are represented by rectangles (big rectangles in Figure 7), which will contain a set of ordered tasks and the available resources. It is the way to locate those elements inside the organization. The following organizational units are the structural decomposition of the hospital: (i) *reception*: patients coming to this unit will be dispatched through the medical units of the hospital; (ii) *general medicine*: diagnostic and simple medical acts are realized in this unit; (iii) *surgery*: patients will be operated in this unit; (iv) *dermatology*: unit involved in every dermatological resource and the performance of the related medical acts; (v) *payment service*.

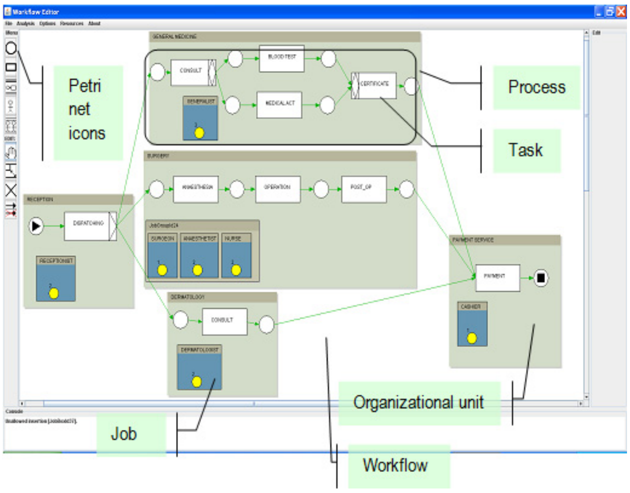
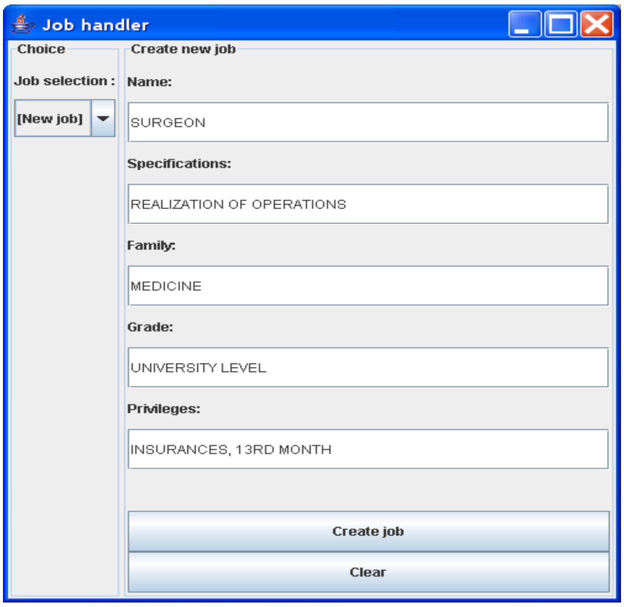


Figure 7. FlowiXML editor.

Who? *Specification of jobs and user stereotypes*. This step consists in the description of all the actors involved in the workflow. For this purpose we define different levels of users, who are the resources that will be in charge of performing the organization work. Jobs are ways to structure the crew of people inside the organization (Figure 8). It involves the complete collection of knowledge and practices needed by a definite human resource to perform a task. Jobs specified in the definition of the case study are the following: Receptionist, Generalist, Surgeon, Anesthetist, Nurse, Dermatologist and Cashier. Once jobs are defined it is possible to incorporate user stereotypes, people able to carry out tasks of a particular job.



Job selection : [New job]

Name: SURGEON

Specifications: REALIZATION OF OPERATIONS

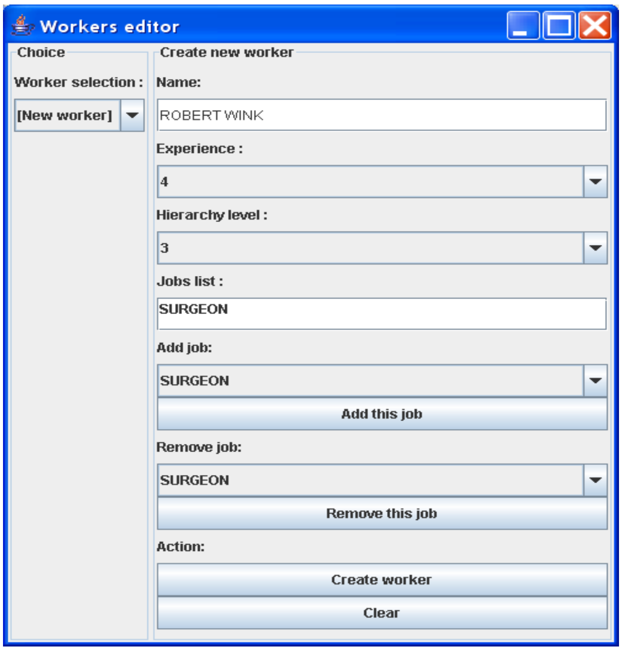
Family: MEDICINE

Grade: UNIVERSITY LEVEL

Privileges: INSURANCES, 13RD MONTH

Figure 8. Job handler editor.

The workers editor (Figure 9) is used for this purpose. Workers are defined in terms of attributes (name, experience, hierarchy level) and the list of jobs they can perform. For instance, we define a user stereotype called Robert Wink, having 4 years experience in the third hierarchy level. He is able to carry out tasks as a generalist and surgeon. A data base of workers can be specified using this editor. Also, it is necessary to assign them a place into the organizational scheme. A user stereotype may be assigned to several organizational units. Hierarchically, a worker being in an organizational unit also belongs to every unit included in the main one. The graphical representation used for the workflow editor is based on a first resource container inside the organizational unit. It allows the workflow designer to group resources. Job boxes are put inside of the main resource box. Each job box is instantiated by user stereotypes able to perform the job of the box. This leads to the kind of representation given in Figure 7 (small rectangles). The organizational unit contains a resource box made of three job boxes. Every job box instantiates user stereotypes of a certain job (there are two surgeons, one anesthetist and one in the given example). This lets managers know which resources are available for execute a task in an organizational unit.



Worker selection : [New worker]

Name: ROBERT WINK

Experience : 4

Hierarchy level : 3

Jobs list : SURGEON

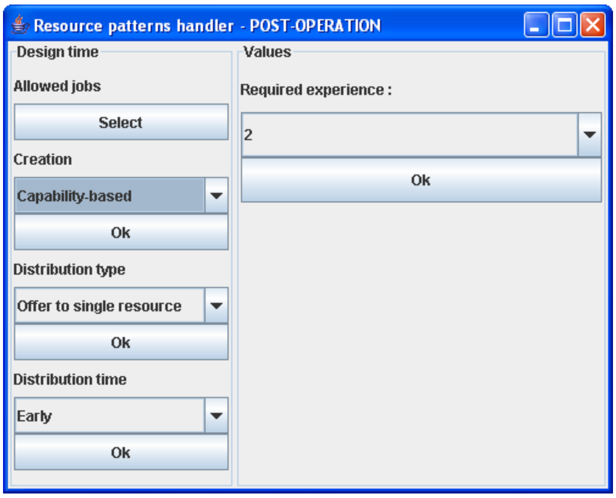
Add job: SURGEON

Remove job: SURGEON

Action:

Figure 9. Workers editor.

Whom? *Defining workflow resource patterns*. It is important to specify who will be in charge of what. For that purpose, we use workflow resource patterns to assign or offer tasks. As, we have already defined jobs and user stereotypes, now we add rules defining the way work will be undertaken. The resource pattern editor (Figure 10) allows the workflow designer to specify resource patterns. At first a list of jobs required to carry out task is specified in the editor. The workflow designer selects one or more jobs allowing a user stereotype to realize the task. For the moment, some workflow resource patterns have been incorporated so that the designer may apply them directly using a predefined UI.



Allowed jobs

Creation: Capability-based

Distribution type: Offer to single resource

Distribution time: Early

Required experience : 2

Figure 10. Workflow resource patterns editor.

How? *Task models specification*. For each process a task model can be specified to describe in detail how the task is performed. By exploiting task model descriptions different scenarios could be conducted. Each scenario represents a particular sequence of actions that can successfully be performed to reach a task goal (Figure 11). We use a plug-in of IdealXML [17].

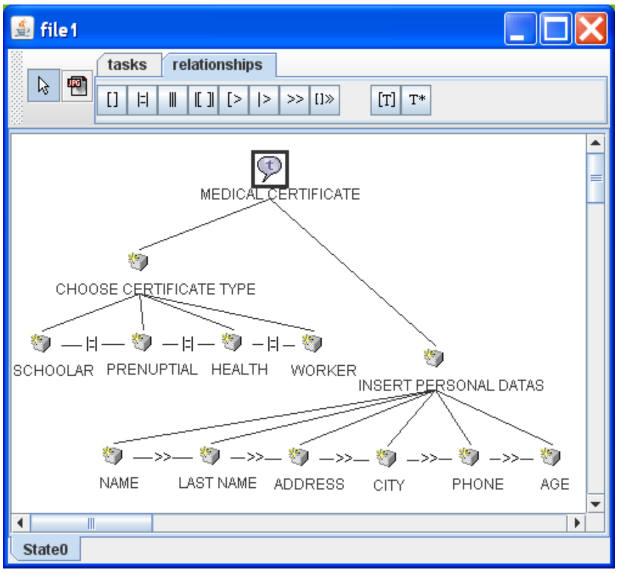
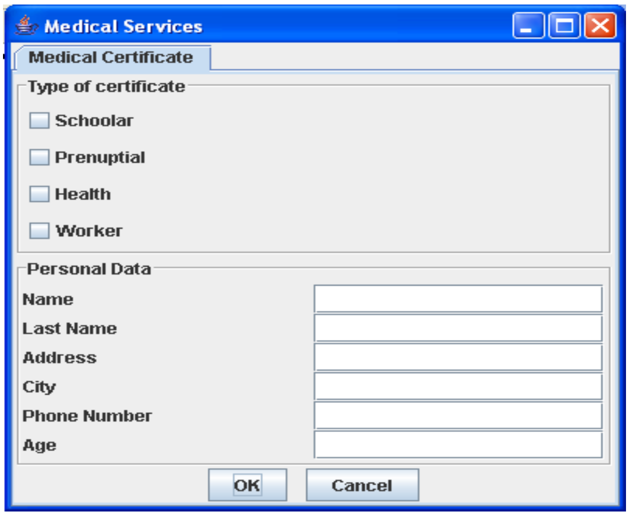


Figure 11. IdealXML editor [17].

Mapping the workflow to UI. Finally we have to deal with the problem of generating the complete UIs set to support all the designed workflow in run-time. This step is achieved by relying on the UsiXML method that progressively moves from a task model to a final user interface. This approach consists of three steps: deriving one or many abstract user interfaces from a task model, deriving one or many concrete user interfaces from each abstract one, and producing the code of the corresponding final UIs.



Medical Services

Medical Certificate

Type of certificate

□ Schoolar

□ Prenuptial

□ Health

□ Worker

Personal Data

Name

Last Name

Address

City

Phone Number

Age

OK Cancel

Figure 12. User interface derived from task model.

To ensure these steps, transformations are encoded as graph transformations performed on the involved models expressed in their graph equivalent. For each step, a graph grammar gathers relevant graph transformations for accomplishing the sub-steps. For instance, applying this method to the task model we obtain its correspondent UI (Figure 12).

CONCLUSION

This paper defined a method for designing UI of workflow information systems where UI are directly derived from a model of the workflow, which is decomposed into processes to end up with tasks. The goal here is not to provide yet another model of workflows, but to identify a meta-model of concepts that are considered fundamental in order to address challenges posed by UI of these workflows. This could achieved by identifying existing related models and consolidate them.

Based on workflow patterns, it is possible to model an entire workflow with high-level mechanisms and automatically generate the workflow specifications and their corresponding UIs. All models are uniformly expressed in the same XML-based specification language so that mappings between models are preserved at design-time and can be exploited at run-time in needed. Then, the different *steps* of the approach have been properly defined based on the underlying models and a *tool* has been developed to support the method enactment. The major benefit of the above method is that all the design knowledge required to progressively move from a workflow specification to its corresponding UIs is expressed in the model and the mapping rules.

The method preserves continuity (all subsequent models are derived from previous ones) and traceability of its enactment (it is possible to trace how a particular workflow is decomposed into processes and tasks, with their corresponding user interfaces). In this way, it is possible to change any level (workflow, process, task, and UI) and to propagate the changes throughout the other levels by navigating through the mappings established at design time. In order to partially support this method, a software tool has been developed in Java 1.5 that supports the graphical editing of the concepts introduced in an integrated way. This method has been so far validated on several real-world case studies. As future work we would like to include workflow analysis to our approach in order to get statistics of the users of the workflow, managing of exceptions (it is important to handle abnormal events that may happen during the execution of a task), and include further implementation experiences, to allow the refinement and refreshment of our approach.

FlowiXML web site

More information, including the tool, a video demo, link to other case studies, can be found at http://www.usixml.org.

Acknowledgements

We acknowledge the support of the ITEA2 Call 3 UsiXML (User Interface extensible Markup Language – http:// www.usixml. org) European project under reference #2008026 and its support by Région Wallonne, the support of the CONACYT program by the Mexican government http://www. conacyt.mx). Thanks to Christophe Lemaigre for the workflow editor and Francisco Montero for IdealXML.

REFERENCES

[1] Brandl, A., Generating User Interfaces for Ad-hoc Workflows. Poster demo at IUI'01. Santa Fe, USA, 2001.

[2] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers 15*, 3 (2003), pp. 289–308.

[3] Casati, F. et al. *WIDE Workflow model and architecture*. Technical Report 96-19, Centre for Telematics and Information Technology (CTIT), University of Twente (1996).

[4] Dumas, M. and ter Hofstede, A. UML Activity Diagrams as a Workflow Specification Language. In M. Gogolla and C. Kobryn (eds.), *Proc. of 4th International Conference on the Unified Modeling Language UML'2001*, pp. 76-90, Toronto, Canada, 2001.

[5] Esposito, D. Getting Started with Microsoft Windows Workflow Foundation: A Developer Walkthrough. September 2005. http://msdn.microsoft.com/winfx/ reference/ workflow.

[6] Guerrero, J., Vanderdonckt, J., and Gonzalez Calleros, J.M. Towards a Multi-User Interaction Meta-Model. Working paper 08/28 ULC, Louvain School of Management. December 2008.

[7] Guerrero, J., Vanderdonckt, J., and Gonzalez Calleros, J.M. FlowiXML: a step towards designing workflow management systems. *Journal of Web Engineering* 4, 2 (2008), pp. 163-182.

[8] Guerrero, J., Vanderdonckt, J., González Calleros, J.M., and Winckler, M. Towards a Library of Workflow User Interface Patterns. In *Proc. of 15th Int. Workshop on Design, Specification, and Verification of Interactive Systems DSV-IS'2008* (Kingston, July 16-18, 2008). Lecture Notes in Computer Sciences, Vol. 5136, Springer, Berlin, 2008, pp. 96-101.

[9] Guerrero-García, J., González-Calleros, J.M., Vanderdonckt, J., and Muñoz-Arteaga, J. A Theoretical Survey of User Interface Description Languages: Preliminary Results. In *Proc. of Joint 4th Latin American Conference on Human-Computer Interaction-7th Latin American Web Congress LA-Web/CLIHC'2009* (Merida, November 9-11, 2009). IEEE Computer Society Press, Los Alamitos, 2009, pp. 36-43.

[10] Helms, J., Schaefer, R., Luyten, K., Vermeulen, J., Abrams, M., Coyette, A., and Vanderdonckt, J. *Human-Centered Engineering with the User Interface Markup Language*. In Seffah, A., Vanderdonckt, J., Desmarais, M. (eds.), "Human-Centered Software Engineering", Chapter 7, HCI Series, Springer, 2009, pp. 141-173.

[11] IBM WebSphere® MQ Workflow. (2006). Retrieved June 2006, from: http://www-306.ibm.com/software/ integration/wmqwf/

[12] Kristiansen, R. and Trætteberg, H. Model-Based User Interface Design in the Context of Workflow Models. In *Proc. of Tamodia'2007*. Springer, Berlin (2007) 227–239.

[13] Limbourg, Q. and Vanderdonckt, J. UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence. In Matera, M., Comai, S. (eds.), "Engineering Advanced Web Applications", Rinton Press, 2004, pp. 325-338.

[14] Luyten, K., Abrams, M., Vanderdonckt, J. and Limbourg, Q. *Developing User Interfaces with XML: Advances on User Interface Description Languages*, Proc. of Advanced Visual Interfaces 2004 Workshop UIXML'2004, Galipoli, 2004.

[15] Luyten, K., Van den Bergh, J., Vandervelpen, Ch., and Coninx, K., Designing distributed user interfaces for ambient intelligent environments using models and simulations. *Computers & Graphics* 30, 5 (2006) pp. 702-713.

[16] Marshak, R.T. Workflow: Applying Automation to Group Process. In Coleman, D. (ed.): Groupware-Collaborative Strategies for Corporate LANs and Intranets. Prentice Hall PTR, 1997, p.p. 143-181.

[17] Montero, F., López-Jaquero, V. IdealXML: An Interaction Design Tool-A Task-Based Approach to User Interfaces Design. In *Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006* (Bucharest, 6-8 June 2006), Chapter 20, Springer-Verlag, Berlin, 2006, pp. 245-252.

[18] Object Management Group: Business process modeling notation specification, final adopted specification dtc/06-02-02 (2006).

[19] Russell, N., ter Hofstede, A.H.M., Edmond, D., & van der Aalst, W.M.P. (2004). Workflow Data Patterns (Tech. Rep. FIT-TR-2004-01). Brisbane, Australia: Queensland University of Technology.

[20] Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., and Edmond, D. Workflow Resource Patterns. In O. Pastor and J. Falcao e Cunha (eds.), *Proc. of 17th Int. Conference on Advanced Information Systems Engineering CAiSE'05*, Vol. 3520 of Lecture Notes in Computer Science (pp. 216-232). Springer-Verlag, Berlin, 2005.

[21] Stavness, N. Supporting Flexible Workflow Process with a Progression Model. Unpublished master's thesis. University of Saskatchewan, Saskatoon, 2005.

[22] van der Aalst, W.M.P. The application of Petri Nets to Workflow Management. Journal of Circuits, Systems, and Computers 8,1 (1998) 21–66.

[23] van der Aalst W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B. and Barros, A.P. Workflow Patterns. *Distributed and Parallel Databases 14*, 3 (July 2003), pp. 5-51.

[24] Vanderdonckt, J. Model-Driven Engineering of User Interfaces: Promises, Successes, and Failures. In *Proc. of 5th Annual Romanian Conf. on Human-Computer Interaction ROCHI'2008* (Iasi, September 18-19, 2008), S. Buraga, I. Juvina (eds.). Matrix ROM, Bucarest, 2008, pp. 1–10.

[25] van Welie M., van der Veer, G.C., and Eliëns, A An Ontology for Task World Models. In *Proc. of 5th International Eurographics Workshop on Design, Specification, and Verification of Interactive Systems DSV-IS'98* (Abingdon, 1998). Springer, Berlin, 1998, pp. 57-70.

[26] Visual Paradigm International Ltd. Business Process Virtual ARCHITECT, User's guide. Retrieved January 2, 2007. From www.visual-paradigm.com/product/bpva/bpvadocuments.jsp

[27] W3C State Chart XML (SCXML), State Machine Notation for Control Abstraction 1.0. W3C Working Draft (2005). http://www.w3.org

[28] Workflow Management Coalition (1999). Workflow Management Coalition Terminology & Glossary. Document Number WFMC-TC-1011. Document Status – Issue 3.0