# How to Describe Workflow Information Systems to Support Business Process

Josefina Guerrero García, Jean Vanderdonckt, Christophe Lemaige, Juan M. González Calleros
[1]*Université catholique de Louvain, Louvain School of Management*
*Place des Doyens, 1 – B-1348 Louvain-la-Neuve (Belgium)*

## Abstract

*This paper addresses a methodology for developing the various user interfaces (UI) of a workflow information system (WIS), which are advocated to automate business processes, following a model-centric approach based on the 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) to 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. For this purpose, workflow is recursively decomposed into processes which are in turn decomposed into tasks. Each task gives rise to a task model whose structure, ordering, and connection with the domain model allows the automated generation of corresponding UIs in a transformational approach.*

## 1. Introduction

Inside organizations, business processes are performed to ensure that work progress towards accomplishment of goals. The organization of work, both within and between companies, is becoming more and more complicated. This is why (computerized) information systems have been developed to support the management of processes and their coordination. Today Information Systems (IS) are part of the organizations with the mission of: 1) improving organizational performance and individual quality work life and 2) providing international leadership and education in the successful management and use of information technology to achieve business objectives [9]. An information system creates value for the organization as an organizational and management solution to challenges posed by the environment. Nevertheless, they have not had the impact expected on the improvement to perform the tasks on the organizations. Workflow is an excellent way to meet this need. Defined as "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [26]. Workflow Information Systems (WIS) cover the application of information technology to business problems. Its primary characteristic is the automation of processes involving combinations of human activities with information technology applications. Owing to the fact that the users of a IS interact with it through its user interfaces (UIs) in the pursuit of organizational goals, flexibility in creating them is therefore important. We will explore a systematic way to define UIs for a WIS. 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 model, hence the task model is suitable for developing UIs.

A short introduction to task and workflow modeling is given in section 2, and a modeling language is provided. Section 3 presents the underlying conceptual model of the framework. Section 4 describes our approach to develop UIs. Section 5 provides a case study. Finally, in section 6 we conclude and give some notes on future work.

## 2. State of the art

The introduction of Workflow Information Systems in organizations has emerged as a major advantage to provide them with a competitive advantage, as a way to trim cost, to automate process, to reduce time. Implemented properly, workflow applications enable companies to reengineer and streamline business processes; for this reason, the interest in workflow systems has grown dramatically over the last years. A number of approaches have been used to model business processes and workflows; those include notations, languages, and software tools. As notations there are Petri Nets [20, 21], Statecharts Diagrams [25, 28], Business Process Modeling Notation (BPMN) [13, 27], and UML Activity Diagrams [4]. Currently, several models and design methods support the development of complex workflow-based applications providing notations for describing rich business process including tool support for designing, for instance: The Progression Model [18], YAWL [23], Microsoft Windows Workflow Foundation (WWF) [5], WebSphere® MQ Workflow (IBM) [7], WIDE [3], ARIS [17] among others. Due to the large amount of existing workflow products, especially all the commercial software that are equipped today with many different capabilities, we came to a point where it is very difficult to compare these capabilities on a common scheme. For this purpose, a collection of workflow patterns has been identified that provide the basis for an in-depth comparison of commercially available workflow systems. Control-flow patterns [22] 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 [15] are aimed at capturing the various ways in which data is represented and used in workflows. Workflow resource patterns [16] correspond to the manner in which tasks are allocated to resources, that is any entity that is capable of achieving some work unit.

All these approaches for workflow modeling involved in some way the task concept. However none of these methods clearly specify the boundaries of a task model and user task which is essential to the development of usable UIs. Examples of task modeling languages are: Hierarchical Task Analysis (HTA)[1], Task Knowledge Structure (TKS) [8], GroupWare Task Analysis (GTA) [24], and ConcurTaskTrees (CTT)[14] which all support designers by hierarchically decomposing tasks, defining objects manipulated and the role responsible for performing the task. The vast number of task modeling notations results in semantic and syntactic differences which are discussed in [11].

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.

Model-based user interface design is intended to assist in designing user interfaces with a more formal computer supported methodology; there are solutions to developing UI that are based in eXtensible Mark-up Language (XML). UsiXML [10, 19] 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 User Interface Description Language (UIDL) to be used in the remainder of this work because of its capabilities of extensiveness, availability, central storage of models, and its transformational approach.

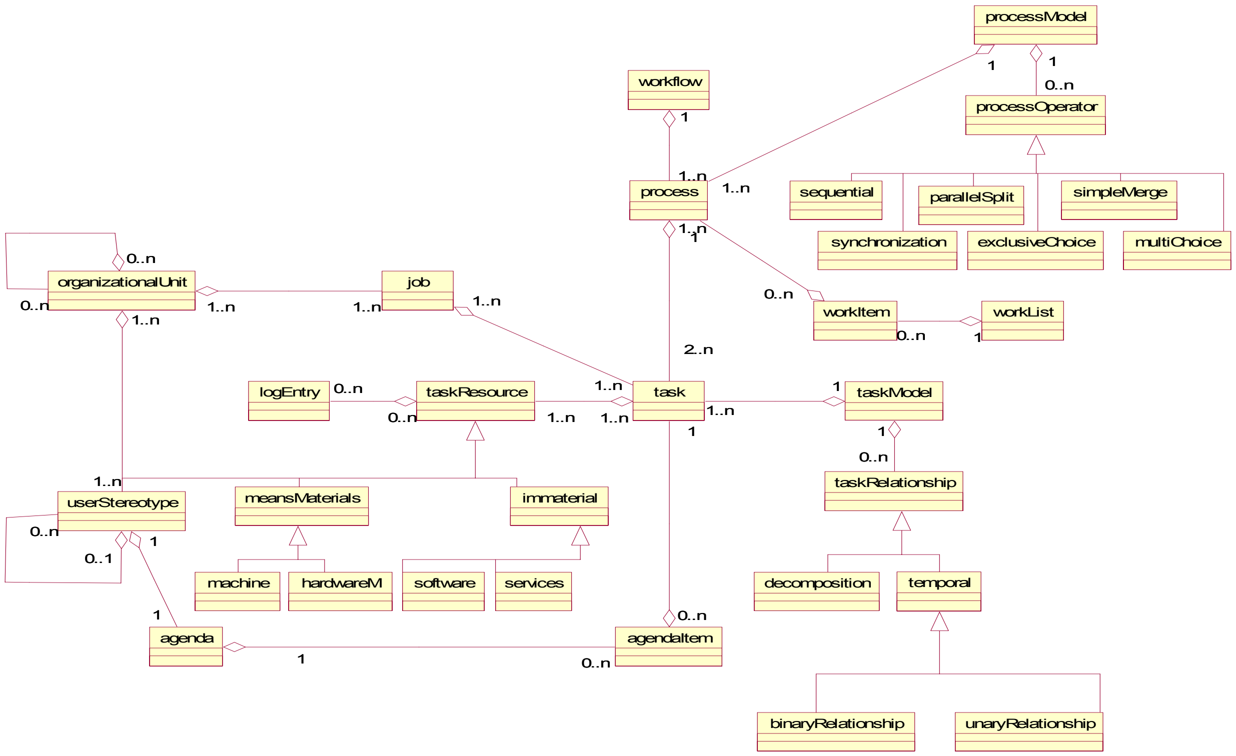


**Figure 1 Partial view of the Workflow Model.**

# 3. Conceptual Modeling of Workflow Information Systems

After reviewing the different approaches to model business process and workflow, we propose a framework (Figure 1) that considers the principal components to model workflow. The intention is to use this model as a base to develop UIs. 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. This kind of analysis is made possible because user tasks are considered from the point of view of the users need for the application and not how to represent the user activity with a particular system.

The underlying conceptual model is composed of: workflow, process, task, and organization models. The workflow model is recursively decomposed into processes which are in turn decomposed into tasks. A process model indicates the ordering of processes in time, space, and resources. Each process gives rise to a process model structured and ordered with process operators. Process operators determine whether the flow of work is sequential, parallel split, exclusive choice or multiple choices, with the corresponding merger operators, synchronization and simple merge. A process model uses these operators to order tasks. The work list allows workflow manager to view and manage the tasks that are assigned to resources. A task model represents a decomposition of tasks into sub-tasks linked with task relationships. We propose an organizational model that is composed of: organizational unit, job, resources, and agenda. An organizational unit is a formal group of people working together with one or more shared goals or objectives. Each task could be supported thanks to a resource model, in which three types of resources can be found: human resources (i.e. a user stereotype), material resources (e.g., hardware, network, machines), and immaterial resources (e.g., software, operating system). Each resource typically fulfills one or more jobs. The agenda displays the tasks in a tabular format, it enables users to organize and manage their personal work.

Transformations are applied in cascade through the workflow layers using a mapping model. In order to support the mapping between the layers, predefined relationships were used: reification, decomposition, isExecutedIn, etc. In this section, only an overview of this concept is provided, for the complete definition of the semantics and the syntax, we refer to [6].

# 4. How to Generate the User Interfaces

To develop UIs, we propose a method that is compliant with the Cameleon Reference Framework [2] for developing multi-target UIs, which is decomposed in four steps (Figure 2):

1) Task and domain modeling (Computing Independent Model in MDA): a model is provided for the end user's task, the domain of activity and, if needed, the context of use (user, computing platform, and environment).
2) Abstract User Interface modeling (Platform Independent Model in MDA): this level describes UIs independently of any interaction modality and any implementation.
3)Concrete User Interface modeling (Platform Specific Model in MDA): this level describes a potential UI after a particular interaction modality has been selected (e.g., graphical, vocal, multimodal). This step is supported by several tools helping designers and developers to edit, build, or sketch a user interface.
4) Final User Interface: this level is reached when the UI code is produced by interpretation or compiled.

As we mention in section 2, UsiXML has been selected as the UIDL. It describes at a high level of abstraction the constituting elements of the UI of an application: widgets, controls, containers, modalities, interaction techniques, etc. UsiXML supports device independence: a UI can be described in a way that remains autonomous with respect to the devices used in the interactions such as mouse, screen, keyboard, voice recognition system, etc. Language provides a means of communication; linguistics—the science of languages—can also be applied to programming languages. By this reason, we introduced the abstract syntax with "*enriched" directed graph*. That is to say an identified, labeled, typed, constrained graph. Finally we presented the stylistics in a graphical representation for the model presented in Figure 1 which relies on icons.

Model-based user interfaces design often starts with a task model that is evolved through an incremental approach to the final UI. For instance, in Figure 2 a task that is no further divided is represented with an Abstract Individual Component (AIC). For each AIC a facet (input, output, control, navigation) and transformational rules are applied. The next step consists in code generation from a CUI to obtain a FUI. For the complete definition of the method, we refer to [19].

As we said before, the aim of this work is to generate the UIs of a workflow information system from its specifications (provided that they address the relevant aspects) and the different constructs that link the tasks and the users together.

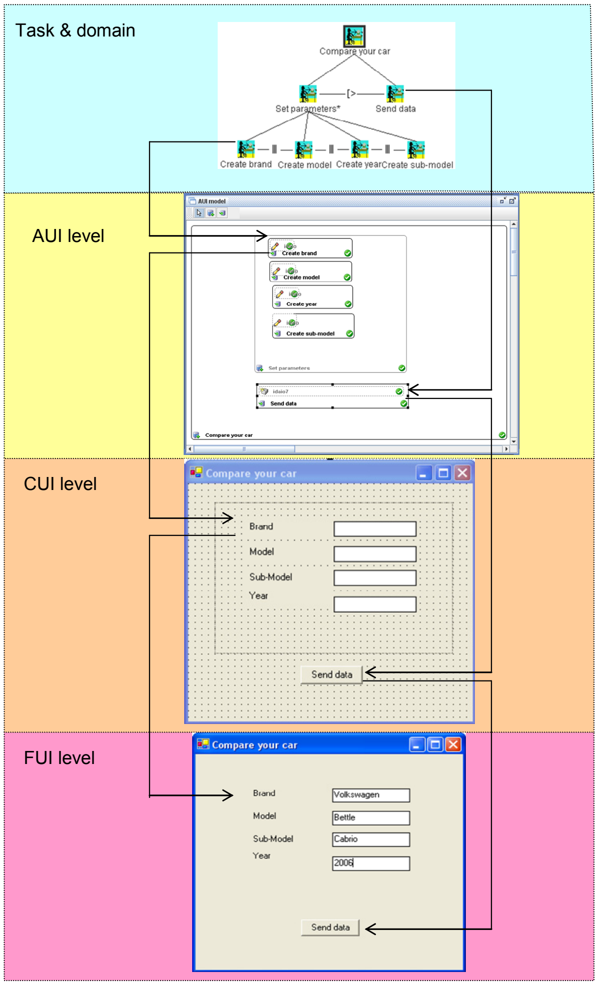


**Figure 2 Steps to derive UIs from task models.**

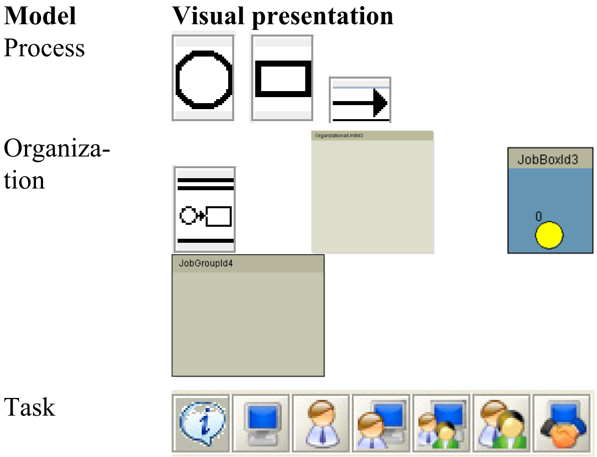


**Figure 3 Stylistic representation**

## 5. Case Study and Tool Support

**Tool support**. In order to support the development of UIs from a workflow model to a task model, a workflow editor has been developed. This editor allows modeling the general workflow defining processes and tasks models, defining organizational units, jobs and resources involved, allocation of tasks to resources, and to manage the flow of tasks. The task model is based on [12].

**Case study**. The case study analyzes how people organize the program of small conferences by using a review tool. To organize a conference it is necessary to identify the tasks that will be done, the jobs and resources which will be involved. These tasks will be assigned to each job taking into account the role that each of them will play in the workflow.

After the specification of the process to follow, the tasks identified are: find the program committee, prepare and distribute the call for paper, install and configure conference tool, find keynotes speakers, submit papers, organize submission and reviewing process, assign papers to reviewers, review papers, give feedback (accept or reject papers), organize discussion to take final decisions, announce results, edit proceedings, compose program, and attend to the conference and present the paper. Table 1 resumes the principal tasks and the jobs in charge of develop them. Also, the identification of resources that are available for doing the work and where they are working, i.e. the organizational unit (Table 2) was specified. Author resources are named as A-1, A-2, and A-3 because we do not know who will be interested in submit a paper.

**Table 1. Tasks and jobs identification.**

| ID | Task | Job | | |
|---|---|---|---|---|
| | | Organizer | Reviewer | Author |
| 1 | Find the program committee | x | | |
| 2 | Prepare the call for paper | x | | |
| 3 | Distribute the call for paper | x | | |
| 4 | Install conference tool | x | | |
| 5 | Configure conference tool | x | | |
| 6 | Find keynotes | x | | |
| 7 | Submit paper | | | x |
| 8 | Organize submission | x | | |
| 9 | Organize reviewing process | x | | |
| 10 | Assign papers to reviewers | x | | |
| 11 | Download paper | | x | |

| | | | | |
|---|---|---|---|---|
| | assigned | | | |
| 12 | Review paper | | x | |
| 13 | Ask reviewers for preferences | x | | |
| 14 | Give feedback (accept or reject papers) | | x | |
| 15 | Organize discussion | x | | |
| 16 | Take final decision | x | | |
| 17 | Announce results | x | | |
| 18 | Prepare final submission | | | x |
| 19 | Edit proceedings | x | | |
| 20 | Compose program | x | | |
| 21 | Attend the conference and present the paper | | | x |
| 22 | Edit final proceeding | x | | |

**Table 2. Resource and organizational unit identification.**

| Resource | Job | Organizational unit |
|---|---|---|
| Chloé Lambin | Organizer | UL |
| Jacques Khelil | Organizer | UL |
| Ellen Martin | Organizer | UL |
| Angela Buker | Organizer | UL |
| Steve Geller | Reviewer | Reviewer's university |
| Rachel Walsh | Reviewer | Reviewer's university |
| Dylan Leitz | Reviewer | Reviewer's university |
| A-1 | Author | Author's university |
| A-2 | Author | Author's university |
| A-3 | Author | Author's university |

Within the workflow editor, we represent each organizational unit, i.e. UL, Reviewer's university, and Author's university (Figure 4).

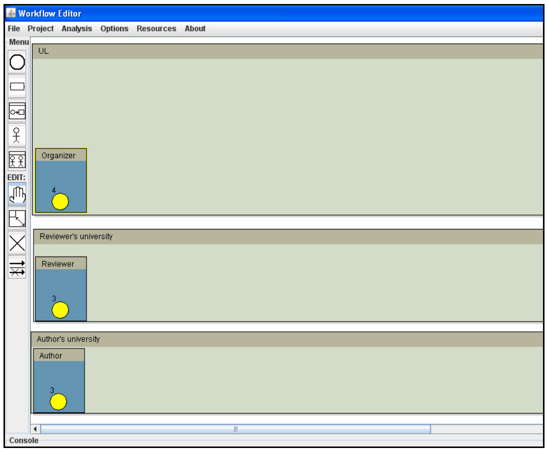
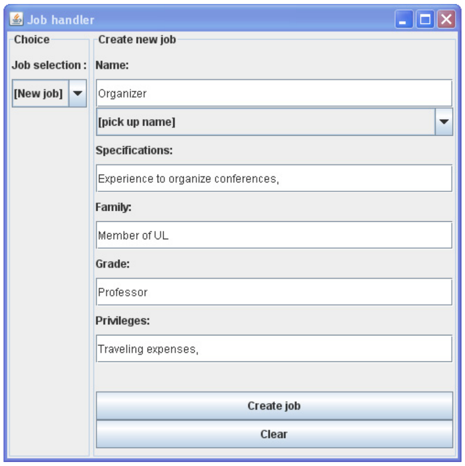


**Figure 4 Specification of organizational units**

After, we specify the different jobs and resources involved in the organization of the conference. It is possible to define some attributes of the job, as the specification, family, grade, and privileges (Figure 5).



Job selection : [New job]

Name: Organizer

[pick up name]

Specifications: Experience to organize conferences,

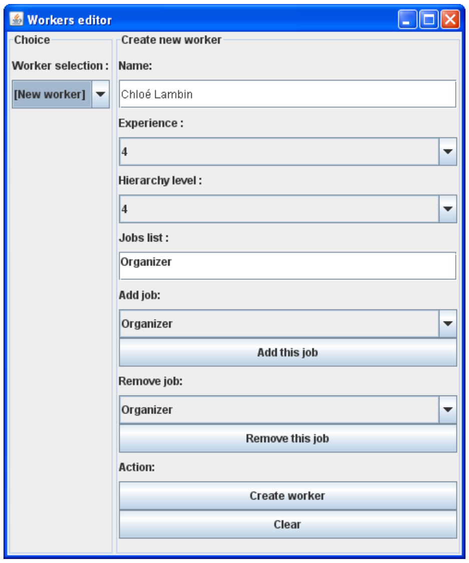
Family: Member of UL

Grade: Professor

Privileges: Traveling expenses,

**Figure 5 Specification of jobs' attributes**

Also, we specify other attributes for the resource (worker): the level of experience, hierarchy level (Figure 6), as requirements that the resource needs to fulfill. The number of resources available for each organizational unit can be specified once the resources have been entered (small dots with a number in Figure 4).



Worker selection : [New worker]

Name: Chloé Lambin

Experience : 4

Hierarchy level : 4

Jobs list : Organizer

Add job: Organizer

Remove job: Organizer

**Figure 6 Specification of resource's attributes**

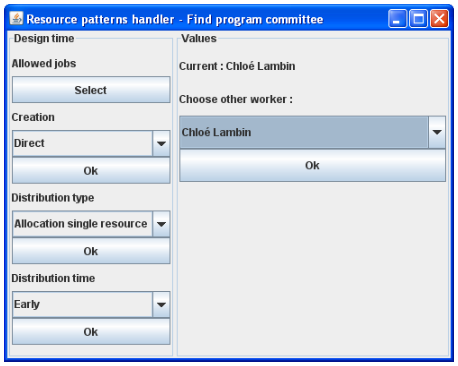
After the identification of tasks, jobs, and resources, tasks are allocated or offered to resources in different ways: direct allocation, delegation, history-based allocation, among others. This assignation (Table 3) is elaborated after the analysis of the characteristics (qualifications, skills, abilities, experience, and hierarchy level) of each resource and considering the requirements of each task. For instance, *Install conference tool* task was assigned to *Ellen Martin* because she is *computer engineering*.

**Table 3. Assigning tasks to resources.**

| Task | Job | Resource | Pattern |
|---|---|---|---|
| Find the program committee | Organizer | Chloé Lambin | Direct allocation |
| Prepare the call for paper | Organizer | Jacques Khelil | Capability based |
| Distribute the call for paper | Organizer | Jacques Khelil | Retain familiar |
| Install conference tool | Organizer | Ellen Martin | Capability based |
| Configure conference tool | Organizer | Ellen Martin | Retain familiar |
| Find keynotes | Organizer | Chloé Lambin | Capability based |
| Submit paper | Author | A-1, A-2, A-3 | Deferred |
| Organize submission | Organizer | Jacques Khelil | History based |
| Organize reviewing process | Organizer | Chloé Lambin | Capability based |
| Assign papers to reviewers | Organizer | Angela Buker | Hierarchy level based |
| Download paper assigned | Reviewer | Steve Geller, Rachel Walsh, Dylan Leitz | Direct allocation |
| Review paper | Reviewer | Steve Geller, Rachel Walsh, Dylan Leitz | Direct allocation |
| Ask reviewers for preferences | Organizer | Chloé Lambin | Retain familiar |
| Give feedback (accept or reject papers) | Reviewer | Steve Geller, Rachel Walsh, Dylan Leitz | Capability based |
| Organize discussion | Organizer | Chloé Lambin | Direct allocation |
| Take final decision | Organizer | Angela Buker | Capability based |
| Announce results | Organizer | Jacques Khelil | Distribution by allocation single resource |
| Prepare final submission | Author | A-1, A-2, A-3 | Deferred |
| Edit proceedings | Organizer | Angela Buker | Capability based |
| Compose program | Organizer | Chloé Lambin | History based |
| Attend the conference and present the paper | Author | A-1, A-2, A-3 | Deferred |
| Edit final proceeding | Organizer | Angela Buker | Capability based |

It is possible to represent those concepts in the workflow editor. First, we select the job and the allocation type. This step allows specifying if the task will be allocated or offered to an individual resource or to a group of resources, and the time that the task is allocated. After, we select the resource (worker) that will develop the task (Figure 7). As we can observe, task Find the program committee is directly allocated to Chloé Lambin, who plays the job of organizer.

In the same way, all the identification of jobs, resources, and tasks allocation is specified.



Resource patterns handler - Find program committee

Design time

Allowed jobs

Select

Creation

Direct

Ok

Distribution type

Allocation single resource

Ok

Distribution time

Early

Ok

Values

Current : Chloé Lambin

Choose other worker :

Chloé Lambin

Ok

**Figure 7 Selecting the resource.**

Once workflow components have been identified and specified on the workflow editor, we can detail in deep the steps to carry out task using task models. To execute the task Submit paper, it is necessary to develop a series of sub-task in order to accomplish the task, i.e. the author needs to log in on web page, the system display the web site where the author fill out his personal data and submit his article, the systems gives a feed back so the author knows if there is an error or not with the file uploaded (Figure 8).

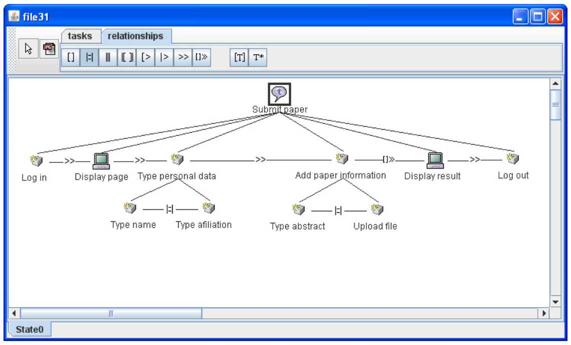
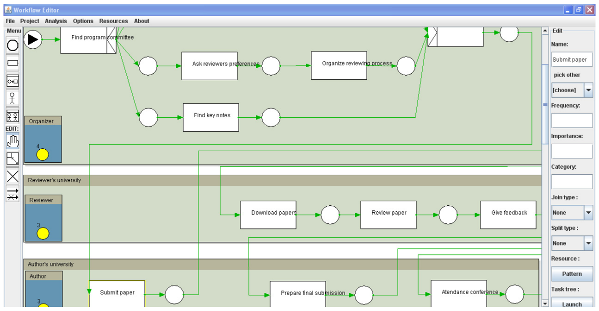


**Figure 8 Representing a task model**

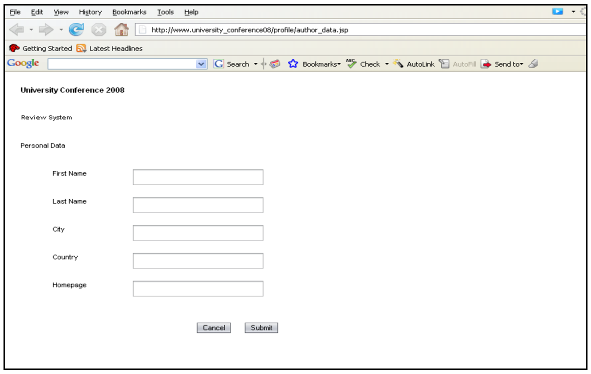
The final workflow model is shown in Figure 9. Now, from task models we derive the UIs of the workflow information system using the method proposed in section 4.



Name: Submit paper
[choose]
Frequency:
Importance:
Category:
Join type : None
Split type : None
Resource :
Task tree :

Figure 9 Workflow to organize a conference

To get the various UIs in this case study, a set of transformation rules were applied [19]. The intention is to generate the UIs for each interactive task involved in the workflow; considering that if the workflow changes, these changes will be reflected in an automatic way in each UI.


File Edit View History Bookmarks Tools Help
Getting Started Latest Headlines
Google
Search Bookmarks Check AutoLink AutoFill Send to
Cancel
Submit

http://www.university_conference08/profile/author_data.jsp

University Conference 2008

Review System

Personal Data

First Name

Last Name

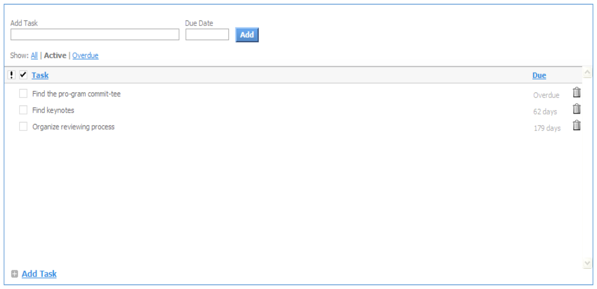
City

Country

Homepage

Figure 10 User interface example

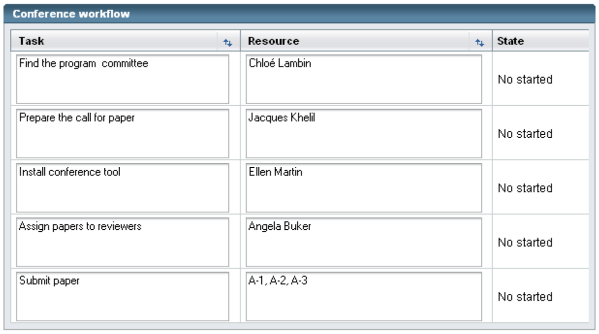
As we mention on section 3, it is necessary to advertise to resources about the tasks that are available or allocated. For instance, tasks assigned to Chloé Lambin, who plays the job of organizer, are listed in her agenda (Figure 11).



| ! ✓ Task | Due |
|---|---|
| □ Find the pro-gram commit-tee | Overdue |
| □ Find keynotes | 62 days |
| □ Organize reviewing process | 179 days |

Figure 11 Agenda

In order to control how the work is flowing, which tasks are in execution or were finished, the workflow manager has his own work list (Figure 12). The purpose of the manager work list is to control how tasks are been developed, if there is a "bottle neck", to know the status of each task, if the deadline is close, if a task is delegated, etc.


Conference workflow

| Task | Resource | State |
|---|---|---|
| Find the program committee | Chloé Lambin | No started |
| Prepare the call for paper | Jacques Khelil | No started |
| Install conference tool | Ellen Martin | No started |
| Assign papers to reviewers | Angela Buker | No started |
| Submit paper | A-1, A-2, A-3 | No started |

Figure 12 Work list

## 6. Conclusion

This paper has addressed a methodology for developing the various user interfaces of a workflow information system, which are advocated to automate business processes, following a model-centric approach based on the requirements and processes of the organization. So far, the focus has been put mostly on the workflow information systems to support business process. For this purpose, a conceptual modeling approach integrates the following concept defined through a meta-model: workflow, process, task, domain, job definition, organizational structure, and resources. These concepts along with their attributes have been integrated in UsiXML V1.8, the last version available of UsiXML today. A transformational approach has been followed to progressively decompose the workflow model into processes which are in turn decomposed into tasks. From each task model, transformational rules were applied in order to generate the different UIs involved in the workflow. A workflow editor-manager 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. As future work, usability guidelines will be applied in the generation of UIs, workflow analysis methods will be taken into account. A case study has been reported and summarized to demonstrate the feasibility of this approach. This method has been so far validated on 4 real-world case studies (e.g., a hospital dept., a triathlon organization, a cycling event, and personalized order of compression stockings over Internet). More information, including a video demo of the software can be found at http://www.usixml.org/index.php?mod=pages&id=40.

**Acknowledgments.** We gratefully thank the support from the SIMILAR network of excellence, supported by the 6th Framework Program of the European Commission, under contract FP6-IST1-2003-507609 (http://www.similar.cc), the Alban program (www. programalban.org) supported by European Commission, and the CONACYT (www.conacyt.mx) program supported by the Mexican government. All information regarding UsiXML is accessible through http://www.usixml.org.

## References

1. Annett, J. *Hierarchical Task Analysis*. The Handbook of Task Analysis for Human-Computer Interaction, Lawrence Erlbaum Associates, Mahwah, 2004. pp. 67-82.
2. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. (2003). *A Unifying Reference Framework for Multi-Target User Interfaces.* Interacting with Computers, vol. 15, no. 3, 289–308, 2003.
3. Casati, F. et al. *WIDE Workflow model and architecture*. Technical Report 96-19, Centre for Telematics and Information Technology (CTIT), University of Twente.
4. Dumas, M. and ter Hofstede, A.: *UML Activity Diagrams as a Workflow Specification Language*. In M. Gogolla and C. Kobryn, editors, Fourth 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* (2005). http://msdn.microsoft.com/winfx/reference/workflow
6. Guerrero, J., Vanderdonckt, J.Gonzalez, J.M., *FlowiXML: a Step towards Designing Workflow Management Systems*, Journal of Web Engineering, 4(2), pp. 163-182, 2008.
7. IBM WebSphere® MQ Workflow. http://www-306.ibm.com/software/integration/wmqwf/
8. Johnson, P., and Johnson, H.: *Task Knowledge Structures: Psychological basis and integration into system desig*n. Acta Psychologica 78, pp 3-26, 1991.
9. Khazanchi D., Munkvold B. E. *Is Information Systems a Science? An Inquiry into the Nature of the Information Systems Discipline.* The DATA 24 BASE for Advances in Information Systems. (Vol. 31, No. 3). Summer, 2000.
10. Limbourg, Q., 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, Paramus, 2004, pp. 325-338.
11. Limbourg, Q. and Vanderdonckt, J., *Comparing Task Models for User Interface Design*, The Handbook of Task Analysis for Human-Computer Interaction, Lawrence Erlbaum Associates, Mahwah, 2004. pp. 135-154.
12. Montero, F., López-Jaquero, V., *IdealXML: An Interaction Design Tool-A Task-Based Approach to User Interfaces Design*, 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.
13. Object Management Group. Business process modeling notation specification, final adopted specification dtc/06-02-02 (2006).
14. Paternò,F. *Model-based design and evaluation of interactive applications.* Applied Computing, Springer. 1999.
15. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Data Patterns. BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven (2004).
16. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M. and Edmond, D. *Workflow Resource Patterns.* In the 17th Conference on Advanced Information Systems Engineering (CAISE'05). Porto, Portugal. 13-17 June.
17. Scheer, A.W., Nüttgens, M. *ARIS Architecture and Reference Models for Business Process Management*. Institut für Wirtschaftsinformatik, Universität des Saarlandes, Im Stadtwald Geb. 14.1, D-66123 Saarbrücken
18. Stavness, N. and Schneider, K.: *Supporting Flexible Business Processes with a Progression Model*. Workshop: Making Model-based UI Design Practical: Usable and Open Methods and Tool, 2004.
19. UsiXML. www.usixml.org
20. van de Aalst, W.M.P.: *The Application of Petri Nets to Workflow Management. Journal of Circuits, Systems, and Computers*. Vol. 8, No. 1, February 1998, pp. 21-66.
21. van der Aalst, W.and van Hee, K., *Workflow Management: Models, Methods, and Systems*. THE MIT Press, Cambridge. 2002.
22. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14, 3 (2003) 5–51.
23. van der Aalst, W.M.P., ter Hofstede, A.H.M.: *YAWL: Yet Another Workflow Language*. Information Systems 30 (2005) 245—275
24. van der Veer, G.C. and van Welie, M. *Groupware Task Analysis*. Tutorial notes for the CHI99 workshop "Task Analysis Meets Prototyping: Towards Seamless UI Development". 16th May 1999, Pittsburgh PA, USA.
25. W3C: State Chart XML (SCXML): *State Machine Notation for Control Abstraction 1.0*. W3C Working Draft 5 July 2005. http://www.w3.org
26. WfMC 2006. *Workflow Management Coalition*. http://www.wfmc.org/
27. White, S. *Business processing modeling notation* (BPMN), version 1.0, http:// http://www.bpmn.org/
28. Wodtke, D., Weikum, G. *A Formal Foundation for Distributed Workflow Execution Based on State Charts*. In Proc. of the 6th International Conference on Database Theory (ICDT '97), Springer-Verlag, LNCS Series, p. 230-246, 1997.