Chapter 6
A Method for Generating Multiplatform User Interfaces for E-Learning Environments

Juan Manuel González Calleros
Université catholique de Louvain, Belgium

Josefina Guerrero García
Université catholique de Louvain, Belgium

Jaime Muñoz Arteaga
Universidad Autónoma de Aguascalientes, México

Jean Vanderdonckt
Université catholique de Louvain, Belgium

Francisco Javier Martínez Ruiz
Université catholique de Louvain, Belgium

ABSTRACT

In this paper the authors present a structured method for automatically generating User Interfaces for e-learning environments. The method starts with a definition of the learning scenario where the different goals, jobs (professor-student/trainer-learner), and tasks are described and stored in a template. After, the description is mapped to FlowiXML, a learning process authoring tool, where graphically trainers or content designers draw the overall process. A learning process is viewed as a workflow and modeled using Petri net notation. From each step in the process model more details are added using user task models; user's activity interacting with a user interface is stored in such diagrams. Then, a transformational method for developing user interfaces of interactive information systems is used that starts from a task model and a domain model to progressively derive a final user interface. This method consists of three steps: deriving one or many abstract user interfaces from the task model, deriving one or many concrete user interfaces from each abstract interface, and producing the code of the final

DOI: 10.4018/978-1-60566-703-4.ch006

user interfaces corresponding to each concrete interface. The models and the transformations of these models are all expressed in UsiXML (User Interface eXtensible Markup Language) and maintained in a model repository that can be accessed by the suite of tools. Developing user interfaces in this way facilitates its automated generation over multiple computing platforms while maintaining portability and consistency between the multiple versions. Our approach is illustrated on an open Learning environment using a case study.

INTRODUCTION

Knowledge acquisition in e-learning environments requires both, individualization of content and social interaction based on learning objects. The learning process links users to domain-specific information sources in collaboration spaces designed for knowledge transfer and knowledge generation. E-learning Communities describe social settings where knowledge can be exchanged effectively. Communities are composed of people who share common interests or needs following a set of rules or policies using computer technologies (Preece, 2000). In recent years, there has been a vast interest in how groups of people work together, and in how collaboration and cooperation might be supported. E-learning communities are formed and exploited by a variety of social and professional groups interacting via the Internet. An E-learning Community is a network of individuals who share a domain of interest about which they communicate online. The participants share the environment, resources, experiences, problems and solutions, tools, and methodologies.

Becoming efficient and stimulating for better and effective learning process using available technology requires a strategy to follow. The strategy must consider at least: design of learning content, design of different ways to present content (textual, graphical or mixed) considering different devices (PDA, mobile, laptop), and defining collaboration during learning process. In this chapter our objective is to describe a method to generate collaborative scenarios with learning objects that can be manipulated in different platforms. We refer to collaborative scenarios as a description of the foreseen interaction between instructors, learners and the system; this information is gathered in a template using MACOBA (Margain, Muñoz, & Álvarez, 2008). Each scenario is viewed as a workflow (depiction of tasks during which documents or information is passed for one participant to another according to a set of rules) that is recursively decomposed into tasks, that could be associated to a learning object (LO); then each task gives rise to a task model whose connection with others models allows the design of user interfaces (UIs) using a transformational approach (Vanderdonckt, 2005). Normally, interaction and communication among e-learning community members take place through a technological interface.

In the reminder of the chapter, we provided a brief background about e-learning environments and the generation of user interfaces, next we present the models involved and the method to generate collaborative multiplatform scenarios exemplifying with a case study. The chapter is wrapped up by summarizing our work, addressing future trends, and deriving conclusions.

BACKGROUND

E-Learning Environments

There is a plethora of computer-assisted e-learning environments/tools (Wainwright, Osterman, Finnerman, & Hill, 2007). The platform of choice

Table 1. Comparing different learning tools

	License	Language	UI Design	Collaborative facilities	Scalability
Claroline	O	PHP	+	+	+ +
Dokeos	F, O	PHP	+	+	+ +
Moodle	F, O	PHP	+	+	+ +
Sakai	F, O	Java	+	+	+ +
Blackboard	C	Java	+	+	+ +
Sloodle	F, O	PHP	+	+	+
ECOOL	F, O	Neutral	+ +	+ +	+ +

for most of the learning environments is the web browser (Bär, Häussge, & Rößling, 2007). Cesarini, Monga, & Tedesco (2004) contribute to have scalable and open architectures among existing solutions. Also, common elements are: tools for creating course material, assessment as well as collaborative tools (forums, emails and chats). These tools achieve the goal imposed by the first requirement because with them we could deliver interaction during and after the lecture, i.e. synchronous and asynchronous learning modes (Li, Lau, Shih, & Li, 2008).

Also, users and jobs are considered in most of the systems but collaboration is neither controlled nor modeled (Wainwright et al.., 2007). Some researchers are exploring the power of workflow modeling in ELE e.g. carrying on the e-learning process with a workflow management engine (Cesarini et al.., 2004) where the system is aware of the different interests of learners. These approaches are near to our work. However, our proposal expands the concept with the integration of User Interface (UI) definition and the modeling of Learning Objects (LOs).

Moodle an ELE to support lectures (Bär et al.., 2007; Dougiamas & Taylor, 2003; Wainwright et al.., 2007) is constrained to a single platform. It is possible to work on different platforms but the system itself is not capable of deliver such changes, so there is a need to program each added platform from scratch. Other possibility to overpass the problem of the extensions is using a Service Oriented Architecture (SOA) as in (Costagliola et al.., 2006) where some functionality is leaved outside the ELE and make it available though web services.

In ELE the process of creation of the UI follows the following approach: there is a general theme and platform, and results are always the same, this means no context adaptation mechanism is included. Major changes could be delivered by plug-ins (also to cover the scalability requirement) but this implies a separated design process and recoding. For instance, Yingling (2006) proposes a migration of Moodle to the mobile domain. The process involves manual changes in order to fit the new platform.

User Interface Development and context adaptation to the platform is considered in ECOOL (Martinez Ruiz, Gonzalez Calleros, Guerrero García, Muñoz Arteaga, & Vanderdonckt, 2008). Since ECOOL posses a meta-definition of the UI in order to provide portability and ease in the updating and migration processes. Besides that ECOOL includes workflow features that are missing or not integral part of other environments. Workflow includes users and jobs managements, working place definition and task specification. From this rich description User Interfaces are derived.

For the sake of simplicity, Table 1 presents only a few of the possible ELE existing in the market. The comparison symbols mean: + supported, ++

Table 2. Comparison of collaborative learning environment design methods

Criteria/Work	(Jonassen & Rohrer-Murphy, 2006)	(McDonald et al., 2005)	(Constance et al., 2002)	(Germán Sánchez et al., 2007)	Current work
Formal specification technique	*Activity theory*	*Conceptual framework*	*PBL*	*State machine*	*Workflow*
Environment	*NonA*	*C-Flow*	*STEP*	*Cated*	*Ecool*
Personalization	+	+	++	-	++
Multiple User interface.	--	+	+	--	++
Reverse engineering	--	+	+	--	++
Customization	+	+	++	+	++
Learning objects	+	++	++	++	++
Multimedia content	+	++	++	+	++
Traceability of collaborative learning	-	--	-	+	++

strongly supported, **-** not supported. The parameters used to compare the ELE correspond partially to main requirements identified. We use the following characteristics identified in (Bär et al.., 2007) to compare them: facilities to interact during and after the lecture (Collaborative facilities); an open architecture which should include the possibility of allow extensions (scalability), including that the system should be able to manage a single course or a whole organization. Support for designing UIs and the type of license, i.e., **C**ommercial, **F**ree software or **O**pen Source. Also the programming language used for adding content.

User Interface Development Methods for E-Learning Environments

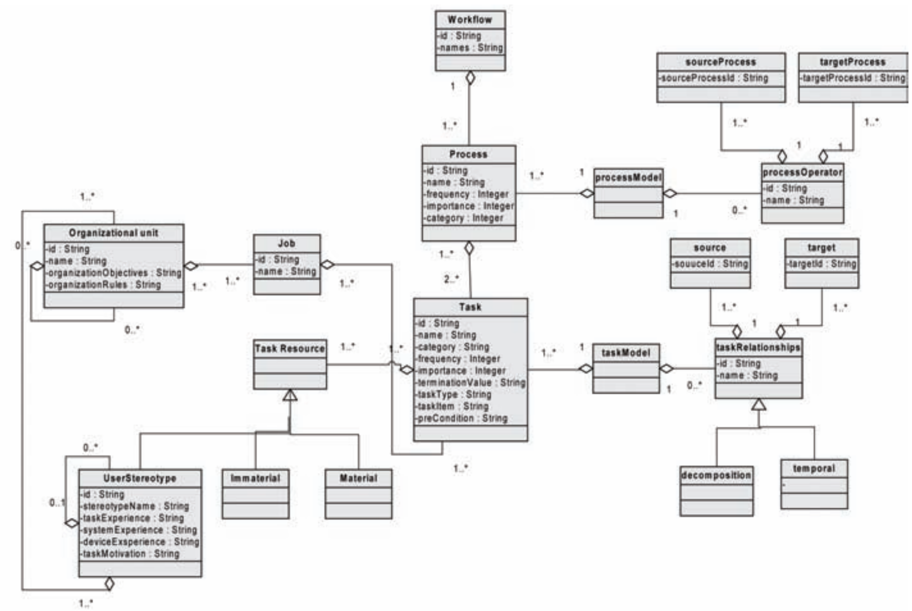
In general, a method for designing and developing a collaborative e-learning environment could use a formal specification technique in order to modeling the evolution of learning process and also it could offer a collaborative learning environment where users can apply and execute some models. In the literature exists several methods to compare with our approach, we observe in Table 2 that there is not a lot methods to take into account all these criteria. Specially, two capital features in the analyzed methods: Customizable interfaces and Personalization. Most of the methods are unable to offer an ELE with UI capabilities similar to ECOOL (Martinez Ruiz et al., 2008).

COMPONENTS OF THE METHOD

There is a global consensus about the components of a User Interface (UI) development method (Vanderdonckt, 2005) which are:

- **Models:** A series of models pertaining to various facets of the UI such as: task, domain, user, presentation, dialog. These models will be defined in the next section and located on a reference framework.
- **Language:** In order to specify different aspects and related models, a specification language is needed that allows designers and developers to exchange, communicate, and share fragments of specifications and that enables tools to operate on these specifications. These models are uniformly and univocally expressed according to a single Specification Language, described in the language model section. The USer

Figure 1. Workflow learning process model



eXtensible Markup Language (UsiXML) (Limbourg & Vanderdonckt, 2004), a User Interface Description Language (UIDL). UsiXML has been selected because of its availability, capacity of expansion.

- **Approach:** An approach refers to the research line or paradigm to be followed by the method. The proposed method Section defines this approach by combination of models and model transformations so as to be compliant with the Model-Driven Architecture (MDA), to support Model-Driven Development (MDD).
- **Tools:** A suite of software engineering tools that supports the designer and the developer during the development life cycle according to the method.

In the following subsections the models and other components implied are introduced.

Learning Process Model

Learning process is described as a workflow, i.e. a progressive series of tasks (Guerrero, Vanderdonckt, González, & Winckler 2008) where the environment, the participants involved and the jobs that each plays, the resources used, and the tasks to be executed are defined. The 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 other models allows the automated generation of their corresponding UI, using a transformational approach. Inside an e-learning community could be a teaching process for the trainers, a learning process for the learners, and an organizational unit for all participants. All these components interact to each other to form an overall learning workflow. In order to have a graphical representation, we can use Petri Nets (van der Aalst & van Hee, 2002) for the specification of processes, and CTT (Paternò, 1999) for the representation of tasks. Figure 1 reproduces a simplified version of the workflow model used, a complete version of it can be found in (Guerrero et al., 2008).

In recent years, there has been a vast interest in how groups of people work together, and in how collaboration and cooperation might be supported. At the same time technology to support working groups is rapidly growing in use, some very important trends are: multiple computing platforms, multiple channels, multiple modalities, multiple environments, and multiple users. Groupware is an application that covers some aspects of the current needs, such as collaboration, communication, coordination, and information sharing. Computer-Supported Cooperative Work (CSCW) is considered a generic term that combines the understanding of the ways in which people work in groups with the available technology of software, hardware, services, and related techniques. Collaborative work is hereby referred to a context of use where several users are worked with coordination, have communication among them, share information in a workspace to achieve a goal. We cover at some level this collaboration of users through the workflow model where jobs of interaction are specified.

Learning Object Model

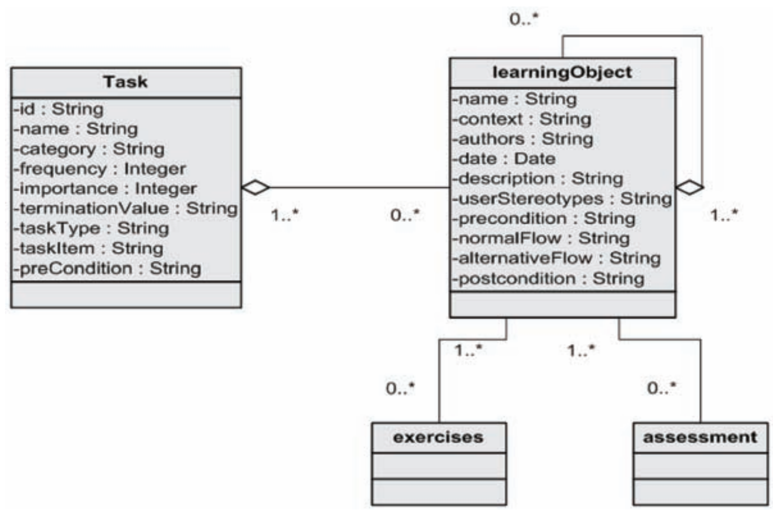
A learning object (LO) is defined as a self-standing, reusable, discrete piece of content broken down into smaller chunks that can be reused in any environment in order to meet an instructional objective. The way LOs are conveyed includes: web pages, PDF documents, video and/or audio content, animations, virtual reality, to mention a few. LOs have been developed to support virtual learning using technology and pedagogical support. These products can be used under any condition or circumstance where the training or the distribution of the knowledge is required; classroom lessons, staff training in the industry, self-learning process, among others.

The structure of a LO showing in Figure 2 could be specified with: a name, a context, authors name, date, brief description, participants involved, pre and post conditions, and normal and/or alternative workflow. The LO model is a simplified version of SCORM standard. A LO can be part or be composed of other LOs. Also, it can be associated to exercises and/or assessments. The LO is part of a task and will be used in a task, this information is relevant when further we explain how a LO is mapped to a UI from a task model.

User Interface Model

Model-based user interface design process starts with a task model that is evolved through an incremental approach to the final UI (Cuppens, Raymaekers & Coninx, 2006). In this way, we selected UsiXML models to design UIs (Vanderdonckt, 2005). UsiXML relies on the Cameleon Reference Framework (Calvary, Coutaz, Thevenin, Limbourg, Bouillon, & Vanderdonckt, 2003). The simplified version, reproduced in Figure 3, structures four development steps: 1) *Task & Concepts* (T&C): describe the various user's tasks to be carried out and the domain-oriented concepts as they are required by these tasks to be performed. 2) *Abstract UI* (AUI): defines abstract containers and individual components, two forms of Abstract Interaction Objects by grouping subtasks according to various criteria, a navigation scheme between the containers and selects abstract individual component for each concept so that they are independent of any modality. An AUI is considered as an abstraction of a Concrete User Interface with respect to interaction modality. At this level, the UI mainly consists of input/output definitions, along with actions that need to be performed on this information. 3) *Concrete UI* (CUI): concretizes an abstract UI for a given context of use into Concrete Interaction Objects (CIOs) so as to define widgets layout and interface navigation. It abstracts a final UI into a UI definition that is independent of any computing platform. Although a CUI makes explicit the Look & Feel of a final UI, it is still a mock-up that runs only within a particular environment. A CUI can also be considered as a reification of an AUI at the upper level and an abstraction of the final UI with respect to the platform. 4) *Final UI* (FUI): is the operational UI i.e. any UI running on a particular

Figure 2. Learning object model



computing platform either by interpretation or by execution.

To support conceptual modeling of UIs and to describe UIs at various levels of abstractions, the following models have been involved:

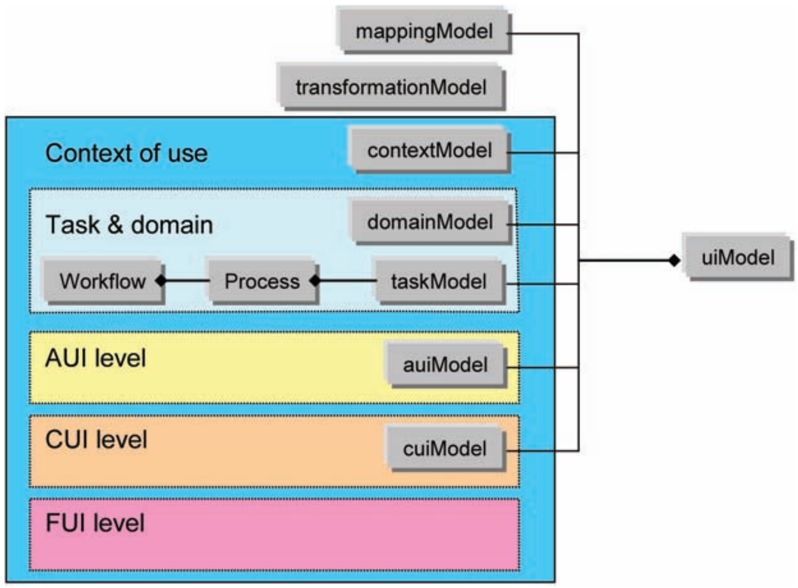
- **domainModel:** is a description of the classes of objects manipulated by a user while interacting with a system.
- **mappingModel:** is a model containing a series of related mappings between models or elements of models.
- **auiModel:** is the model describing the UI at the abstract level as previously defined.
- **cuiModel:** is the model describing the UI at the concrete level as previously defined.
- **transformationModel:** Graph Transformation (GT) techniques were chosen to formalize explicit transformations between any pair of models, except from the FUI level.
- **contextModel:** is a model describing the three aspects of a context of use in which a end user is carrying out an interactive task with a specific computing platform in a given surrounding environment. Consequently, a context model consists of a user model, a platform model, and an environment model. This model plays and important job for multiplatform development and UI adaptation.
- **uiModel:** is the topmost superclass containing common features shared by all component models of a UI.

Language

Designing UIs to display Learning Objects requires a User Interface Description Language (UIDL), which consists of a high-level computer language for describing characteristics of interest of a UI with respect to the rest of an interactive application; it helps define user interfaces linguistically with a general trend to do so in an XML-complaint way. UsiXML, which stands for

USer Interface eXtensible Markup Language (Limbourg & Vanderdonckt, 2004; Vanderdonckt, 2005), has been selected as the UIDL because of its capabilities, such as: extensiveness, availability, central storage of models, and its transformational approach. Also, the graceful degradation of UI (Florins, Montero, Vanderdonckt, & Michotte, 2006) is a feature relevant to this project in particular when adapting the UIs to mobile devices.

Figure 3. UsiXML and Cameleon reference framework



UsiXML supports platform independence: a UI can be described in a way that remains autonomous with respect to the various existing platforms, such as: mobile phone, Pocket PC, Tablet PC, kiosk, laptop, desktop, and wall screen.

For LO we use a simplified version of the Sharable Content Object Reference Model (SCORM), which offers a set of specifications for defining e-learning material. The standards define the communication scheme and how to create packages in the form of zip files. It is important to point out that SCORM is not attached to any instructional style, pedagogical methods of appearance (this is delegated to the ELE). The SCORM API includes a set of methods that can be implemented in any ELE. Additionally, this standard is scalable. Their core model is composed of three aspects: the content (which describes the content), the meta-data and packing. These models are capable of dividing the content into units called Sharable Content Objects (SCOs) and assets. These last are the format for representing media, text, images, sound, and web pages among others (the resources needed to show information to the student). The use of, partial or extended, standards such as SCORM is part of all ELEs, for instance Hard SCORM (Li et al., 2008).

Approach

MDA has been applied to many kinds of business and academic problems and integrated with a wide array of other common computing technologies, including the area of UIs. In MDA, a systematic method is recommended to drive the development life cycle to guarantee some form of quality of the resulting software system (Vanderdonckt, 2005). The transformational development of UI finds its root motivations in the concept of heterogeneousness. In this case the heterogeneousness concerns the variety of contexts of use (referred as a triple <user, computing platform, physical environment>) for which a UI is designed. This heterogeneousness stresses the need for abstractions able to factor out details relevant to specific contexts. From these abstractions, it is possible to obtain context specific representations by progressive refinements. The advantage of accessing to such representations is to be able to reason on one single model and obtain many different UIs.

To cope with the ever increasing diversity of markup languages, programming languages, tool kits, and interface development environments, UsiXML proposes a conceptual modeling of user interfaces (Vanderdonckt, 2005). The conceptual framework was created for specifying, designing, and developing user interfaces at a level of abstraction that is higher than the level where code is merely manipulated. For this purpose, a complete environment has been created based on conceptual modeling of user interfaces of information systems structured around three axes: the models that characterize a user interface from the end user's viewpoint and the specification language that allows designers to specify such interfaces, the method for developing interfaces in forward, reverse, and lateral engineering based on these models, and a suite of tools that support designers in applying the method based on the models. This environment is compatible with the MDA recommendations in the sense that all models adhere to the principle of separation of concerns and are based on model transformation between the MDA levels. The models and the transformations of these models are all expressed in UsiXML (User Interface eXtensible Markup Language) and maintained in a model repository that can be accessed by the suite of tools. Thanks to this environment, it is possible to quickly develop and deploy a wide array of user interfaces for different computing platforms, for different interaction modalities, for different markup and programming languages, and for various contexts of use.

Model-Driven Development of User Interfaces for E-Learning Environment

Considering that scenarios are a description of the foreseen interaction between instructors, learners and the system, they capture the context in which they occur, the process and associated data, and the order of executed tasks; our proposed method starts by defining them in a formal definition using a workflow, i.e., a series of tasks, to specify learning process. In this step we can indicate the resources involved, the organizational unit where they work, how and in which order tasks will be executed (using task model); after, we associate the LO which are stored in a repository and were defined using MACOBA methodology (Margain et al.,2008), finally the whole set of UIs for the ELE is generated. The multiplatform feature is then also available by relying on UsiXML existing tools. In the following subsections these steps are introduced. (see Figure 4)

Specification of Learning Process

An e-learning environment could be viewed as a social space where multiple users produce and share information. Each user has a job (the total collection of tasks, duties, and responsibilities assigned to one or more users) and has a hierarchical position inside the social space. Between the principal users are: teachers, learners, domain experts, and managers. Inside of learning spaces users interact in order to learn, so, users could work as groups in cooperative, collaborative or competitive way. For instance: Dominique Stuart (user), a teacher (job) in computer science faculty (organizational unit), has 10 learners (users/job) which need to work in the resolution of simple equations (task). In order to facilitate the task and integrate all learners, she decides that learners will work in pairs (groups) and the couple that resolves the equations in less time (competition) wins an extra point.

Relying on (Guerrero, Vanderdonckt, González, & Winckler, 2008) learning process is described.

The scenario is described in a complete textual statement describing the learning process to be addressed. This is assumed to be capture using MACOBA template (see Table 4). This textual scenario is aimed to identify elements that are relevant for building a first version of models that will be further exploited in a model-driven engineering method of UIs.

Figure 4. Proposed method

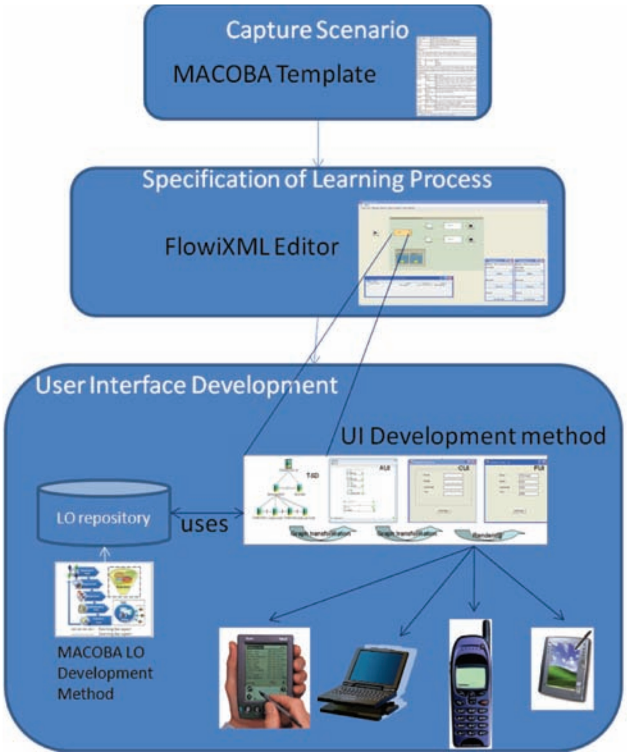
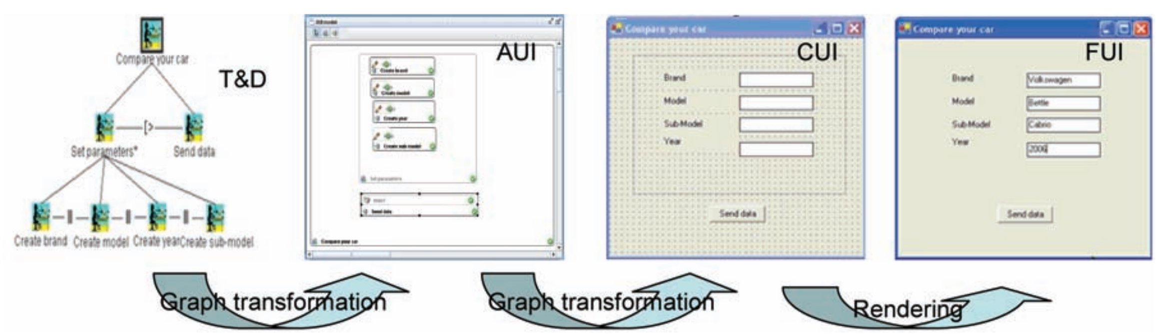


Figure 5. Designing UIs with UsiXML



1. A workflow model is developed using Petri Nets to represent graphically the series of tasks that are involved in the collaborative scenario.
2. After the identification of the task, tasks models are designing using CTT.
3. Defining resources and their jobs in the organization it is an important aspect that we consider. It is not just about identifying different jobs (trainer, learner) and resources involved in the process but also, attributes of the job (specification, family, grade,

Table 3. Main laws of classical mechanics course

Table of Content
The first law
The second law
The third law
The Universal gravitation law
The free fall

privileges) and resource (level of experience, hierarchy level) that can lead to further define who is capable to perform a determined task. After defining tasks, jobs, and resources, tasks are allocated or offered to resources in different ways: direct allocation, delegation, history-based allocation, among others.

User Interfaces Development Method

As we mention before, user interface design processes starts with a task model that is evolved through an incremental approach to the final UI. Figure 6 shows the four levels that are involved in the design of a UI using UsiXML, i.e. the task model, the abstract UI, the concrete UI, and the final UI.

For instance, 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

Table 4. MACOBA template to capture textual scenarios of collaborative learning

Pattern Name		**Collaborative learning**
Context		Classical mechanics, Physic and Mathematic
Authors:		Jaime Muñoz Arteaga and Lourdes Margain
Date:		10/08/2008
Description: This use case describes the collaborative learning with learners they work in groups of 2 or 3 learners who collaborate and they could interact through collaborative tools. This use case covers the curricula and allows access in the case of use; learners are located in different geographic places.		
Actors: Student Teacher	No. Actor 1 2	Jobs: Student Teacher
Preconditions: The Students are connected to internet from different places and collaborate between colleagues throughout the communication service of the learning environment.		
Normal Flow		
Jobs	**Case**	**Description**
Student	Initial exploration (I)	Browsing through the screens which will be explained through text, images, audio and video. These elements are defined as a learning object called “bodies in free fall Newtonian law”.
Student	Auto-evaluation (II)	Students answer the quiz and get feedback from learning object. The quiz also gives the number of successes achieved.
Student, Teacher	Forum Discussions (III)	Teacher listen the Student’s opinions and answer their doubts about bodies in free fall law.
Student	Develop report (IV)	Develop a report about the knowledge learned.
Teacher	Rate Results (V)	To analyze the strengths and weaknesses obtained from the Student and provide suitably qualified.
Teacher, Student	Feedback (VI)	Teacher gives feedback to the Student’s strengths and weaknesses notify them by mail
Alternative Flow: None		
Pos conditions: The collaborative forum has been enabled.		

Figure 6. Rendering of the e-learning environment Icampus on different platforms

applied. The next step consists in code generation from CUI to obtain a FUI. For the complete definition of the method, we refer to (Limbourg & Vanderdonckt, 2004).

From a task model specification it is possible to derive as many UIs as devices have been specified in UsiXML. Figure 6 illustrates the rendering of the E-Learning Environment of icampus (http://www.icampus.ucl.ac.be/) on Mobile devices and a PDA. UsiXML approach covers the multi device and multiplatform support. The process is as follows

The MACOBA methodology (Margain et al., 2008) offers a framework to integrate the future of the collaborative teaching-leaning process based on patterns, in particular for designers. To construct collaborative learning objects with constructed by patterns MACOBA offers four basic levels and one final evaluation (Figure 7):

- **Requirement level:** teachers increase their experience with the planning process. The patterns in this level are guidelines for teachers (instructional designers).
- **Analysis level:** The analysis level implements UML as an innovative way of reusing the use cases patterns and sequence diagrams patterns for collaborative learning.
- **Design and develop level:** the technological designer personalizes the frames patterns and selects communication services as wiki, forum, and chat.
- **Implantation level:** responds to the learning scenario organization. In this level the elements are the activities, jobs, sequences. This level is bases on IMS-LD Model incorporating structure and collaboration activities.
- **The evaluation level:** considers a whole process review about the correct patterns application because is necessary to be sure about the application of collaborative process.

We rely on the existing set of LO resulting of MACOBA approach that can be found at http://ingsw.ccbas.uaa.mx/repo/galeria/. It is out of scope of this work to define LO we assume that the existing repository fulfills our needs. Even more, any new LO can be added following the MACOBA methodology.

TOOL SUPPORT

We designed the ECOOL architecture (Martínez et al., 2008) to support the method. It is composed mainly of two parts (Figure 8): server side and client side The first part includes an ELE composed of seven modules: the LO management module (in charge of LO modification), the assessment engine (in charge of recovering evaluations in order to present them to the user), the translation engine for transforming our definition from/

Figure 7. MACOBA methodology

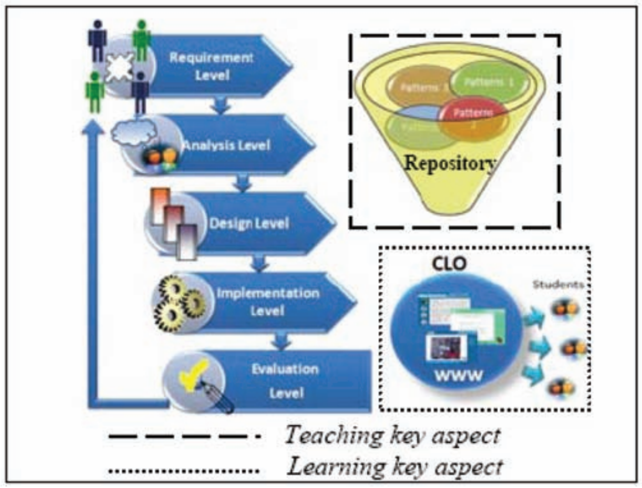
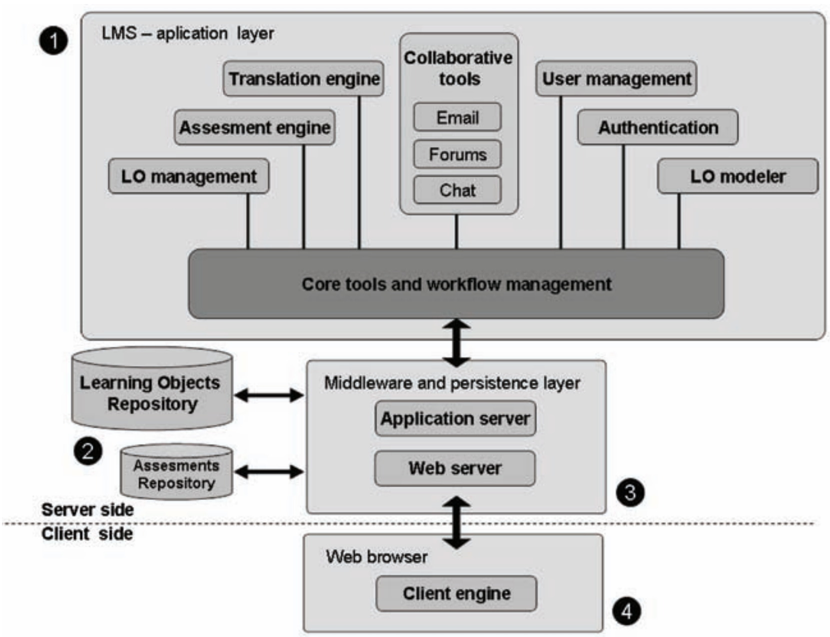


Figure 8. ECOOL architecture



to known standards as well as including rules to produce a LO from the definition of a task to some specific UI, collaborative tools module (Email, forum and chat), the security module (in charge of authentication issues) and finally a designer tool. All these modules are in communication and under the authority of a major component: the workflow management tool. Indeed, this component will deliver the other part of the collaborative process, the modeling and control of the flow of all the activities in the E-Learning environment.

Also, this part deals with the data base control,

Figure 9. FlowiXML learning process editor

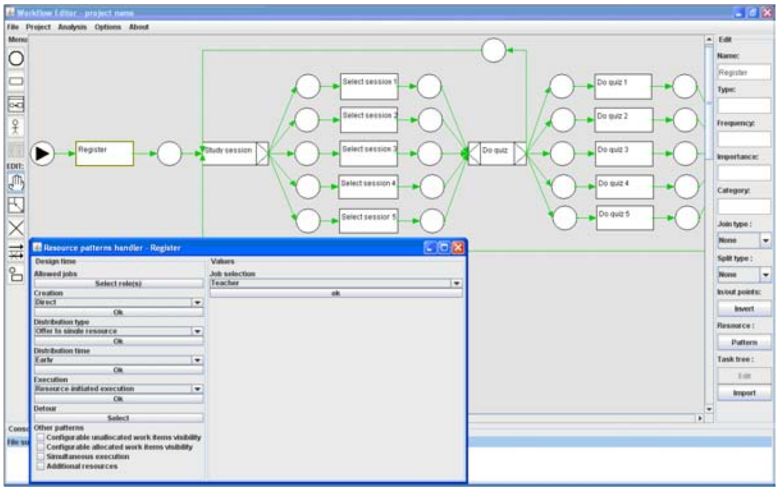
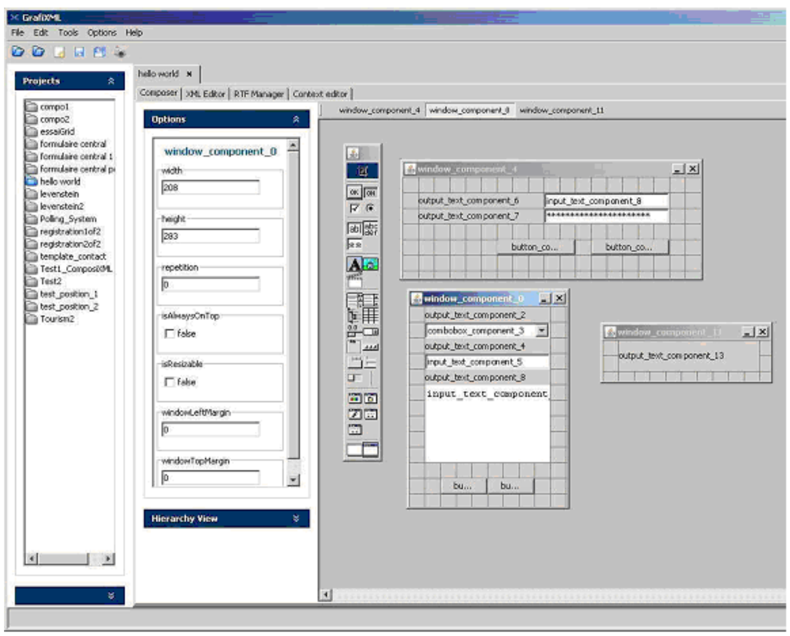


Figure 10. A multi-target user interface



the management, and the application server (Figure 8 section 3). This section corresponds to any typical middleware architecture. The only novelty is the strict separation between the LOs and the assessment repositories (Figure 8 section 2). The second part, i.e. client side, is where after the first admission to the system, it would be mandatory to download a client engine/plug-ins to deal with the virtual renders and extra capabilities provided by the 3D world (González Calleros, Vanderdonckt, &

Arteaga, 2007), all models supported in UsiXML can be a dynamic way to present LOs.

In order to support the development of UIs from a workflow model to a task model, a workflow editor has been developed, Figure 9 (Guerrero et al., 2008). 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.

Finally, if a GUI should run in many different contexts of use, then alternative GUI designs should be specified and added for each new context of use, thus leading to specifying a multi-target UI. GrafiXML (Michotte & Vanderdonckt 2008) an original user interface builder in that it enables designers and developers to design several UIs simultaneously for multiple contexts of use, i.e. for many users, platforms, and environments. For this purpose, it maintains coordination between three representations: an internal representation consisting of specifications in USer Interface eXtensible Markup Language (UsiXML), an external representation consisting of the interface preview, and a conceptual representation consisting of a user interface model. GrafiXML is an intelligent UI builder in that it maintains model consistency between these representations through a set of mappings based on an UI ontology. Thanks to this mechanism, GrafiXML provides a unique set of features for supporting designing interfaces for multiple targets. (see Figure 10)

CASE STUDY

In order to illustrate our approach this section presents a case study for the collaborative learning about laws of bodies in motion. We consider a course to study the laws of classical mechanics (see Table 3).

Each one of the topics of laws of bodies in motion will be encapsulated as a LO and the order of presentation to learners is defined through a workflow. Here the workflow is used to specify the collaborative learning about laws of bodies in motion in order to generate the graphical UI for our learning environment. With the workflow notations is possible specify different mechanisms to establish for example what learning objects are required to the scenario and/or what is the sequences of presentation of these objects in function of learners task.

Capture of the Scenario

For this course the trainer could propose different activities in group in order to teach this law to the

Figure 11. Use case of collaborative learning

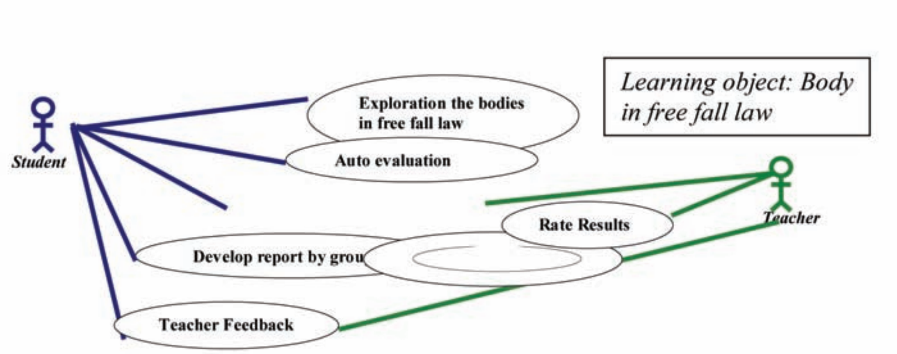
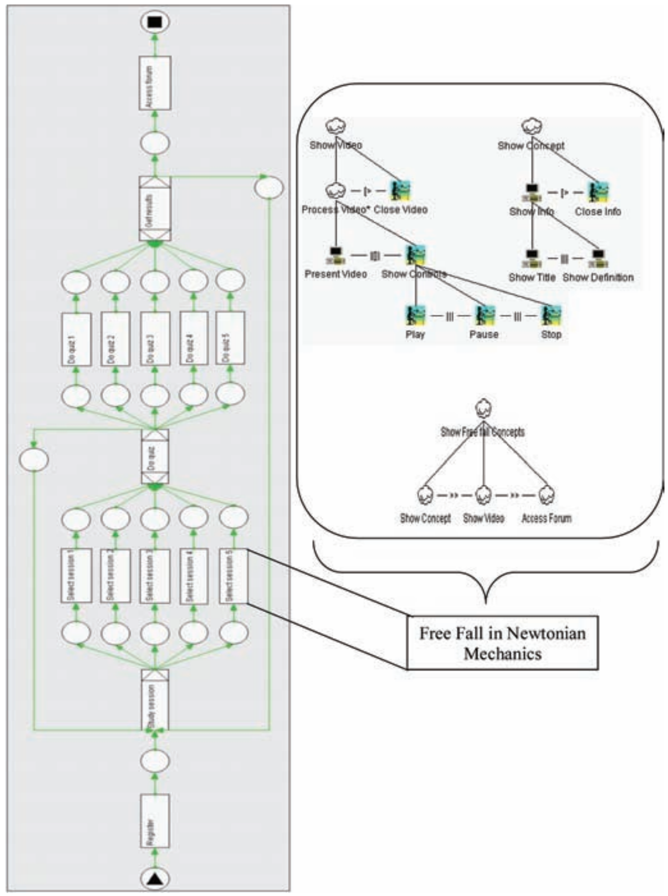


Figure 12. The course of mechanics modeled using petri nets in order to describe its workflow



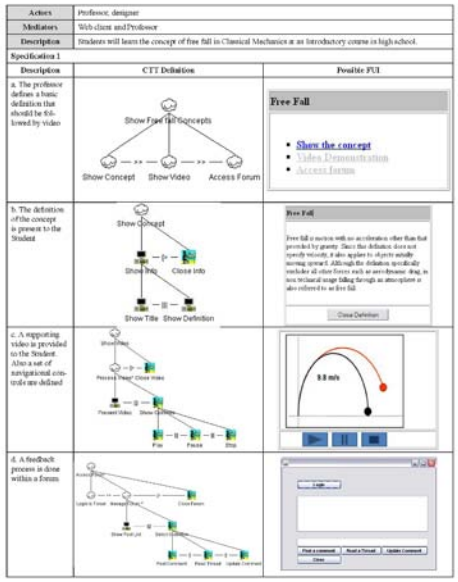
learners in a collaborative manner. These activities are specified with educative analyze pattern proposed in the MACOBA methodology (Margain et al., 2008), as follow in the Figure 11 and Table 4. According to the use case of Figure 11, every learner starts exploring the learning object of bodies in free fall law studying its concepts and interacting with different simulations for this law. An auto evaluation could be taken for learners and be ready to discuss between them and with their teacher throughout a forum web service.

Specification of Learning Process

The pattern for this use case describes the collaborative learning of bodies in free fall law. This pattern specifies that there are users which have the job of learners and other one as teacher. Also we have included a collaborative task that requires a group of three learners who will collaborate in a communication service (forum) see Figure 12 section a. Next, the designer has three options: Define new LOs (to fulfill the learning goal), Modify existing ones and reuse the ones stored in the repository. For this example, the first possibility is selected.

In Figure 12 we could see a list of the LOs proposed by the teacher to cover the laws of bodies in motion. For the sake of simplicity only one is selected: bodies in free fall Newtonian law. The objective is to present the concept of free fall to learners using a textual description followed by a

Figure 13. Table depicting the description of the design of a LO for introduce a concept



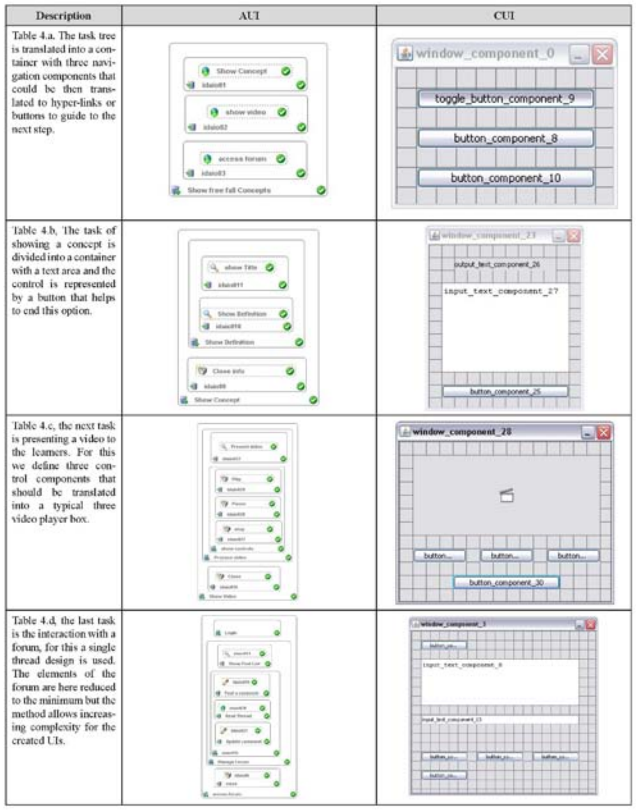
video presentation. So, we model the workflow, indicating which participants will be involved, and specifying the different tasks that are necessary to execute. The tasks will be grouped in task trees which will be translated through the successive process of translation into a FUI as in Figure 5. Here the resulting AUIs and CUIs of the example are shown in order to clarify the process and understand the translation process. For the sake of simplicity, we are only presenting some branches of the Petri Net. In Figure 13 and Figure 14 the development process is described.

According to the learning pattern in Table 4, certain tasks are proposed, they are numbered from one to six using roman numerals. Then, we could see expressed them in a more accurate way. The process is straightforward: for each case defined by the pattern, the professor and the designer have to introduce the needed tasks. For instance, the first case is the exploration of the available material as text, videos, and Web pages; and this process is dosed in a number of sessions that is specified by the professor. Next, learners are invited to evaluate themselves through quizzes and this is done in order to fulfill the second case of the pattern. The following step is discuss between other learners and the professor for resolving doubts and discuss hard topics or problems with the learned concepts, as we have seen until here, a third set of states are defined in our Petri net to model the interaction with the professor in a forum. Finally, the rest of cases are not included since they follow the same principle.

User Interfaces Development

The next step is model each one of the nodes of our workflow in order to define the neutral definition of the UI as in Figure 12. Since this process is the main goal of the method, it was presented before. Our method reminds silence in most of the decisions of the way to create the workflow. This is

Figure 14. Table depicting the possible translations of the tasks tree definitions



intentionally to avoid any problem of portability or assumption of a specific platform.

These task trees should be translated through the successive process of translation into a FUI as in Figure 5. Here the resulting AUIs and CUIs of the example are shown in order to clarify the process and understand the translation process. Again for the sake of simplicity, we are only presenting some branches of the Petri Net. In Figure 13 and Figure 14 the development process is described.

In this section we illustrate the method for developing Multiplatform User Interfaces for E-learning environments based on scenarios description captured in MACOBA template. The feasibility of this approach has been tested and validated.

FUTURE TRENDS

Until recently, e-learning environments, as well as their supporting UIs, have been condemned to stay pre-defined and fixed along most dimensions. In the near future, we expect that these constraints will be relaxed progressively so as to give rise to a new series of open questions including, but not limited to:

- **Platform variation:** As a consequence of the previous variation or independently of that, the worker may want to use another computing platform at run-time because it is more convenient for her. Therefore, the supporting UIs should support the same task on different computing platforms and accommodate their variations.

- **Ad-hoc adaptation of the UI:** When referring to change in the platform there is a need to define what information is relevant to be displayed. As the problem is not just about rendering on a small screen but also to keep the UI with the LO usable.
- **The migration of UIs:** Related to the previous tasks while referring to collaboration among any worker involved in the process there might be scenarios in which splitting the UI in several pieces is necessary but not just splitting but also migrate it to different devices. For instance, imagine a set of learners working on an assessment related to the lecture of the day they all could take a subset of questions migrate them to their mobiles and then put them back and present the UI with the whole assessment.
- **More Dynamic Content:** The use of virtual reality has been explored and successfully used for lectures. However, to present true virtual content depicting LOs has not been integrated in any ELE. For instance, many virtual reality models exist of the Human Body; this might be useful in an anatomy course.
- **Run-time adaptation to the context:** While our method serves to design UI that will be later run in different platforms, it would be interesting to investigate how to adapt to the context in run-time without using a predefined UI but more an automatic transformation for any platform.

CONCLUSION

An alternative method for the automatic generation of Collaborative Multiplatform Scenarios with Interactive LOs has been introduced. The overall learning process is seen as a workflow. This approach introduces a flow control that allows tackling at the same time, the problem of divergence in individual learning and the definition of the learning process in terms of collaboration agents and processes i.e. giving the system the capacity of managing collaboration between learners/trainers, as well as, defining in an explicit way the network and control points of collaboration. Other innovation is the modeling of the interaction of the LOs which includes the introduction of a meta-description (in UsiXML) that is going to aid in the process of generation of multiple UIs to be spread over multiple platforms. Instead of create a new environment we are going to work with one of the most known ones: Claroline (Docq, Lebrun, Smidts, 2007). The process of deliver a modify version is on its way. The next step is to gather all above concepts into an ELE framework. The idea of building learning spaces is that users interact intensively through them and share information in order to learn, then the users could work as groups in cooperative or collaborative way. The set of tools that are used in the method were described shortly because they have been widely described somewhere else. Finally a case study has been presented to illustrate our method.

RESOURCES

All resources related to UsiXML, FlowiXML can be found at: http://www.usixml.org. On this web page software can be downloaded, along with its user's manual, and case studies with examples.

ACKNOWLEDGMENT

We gratefully acknowledge the support of the CONACYT program (www.conacyt.mx) supported by the Mexican government and the SIMILAR network of excellence (http://www.similar.cc), the European research task force creating human-machine interfaces similar to human-human communication of the European Sixth Framework Programme (FP6-2002-IST1-507609).

REFERENCES

Bär, H., Häussge, G., & Rößling, G. (2007). An integrated system for interaction support in lectures. *In Proceedings of the 12th Annual SIGCSE Conference on innovation and Technology in Computer Science Education ITiCSE '07* (pp. 281- 285). New York: ACM.

Birchfield, D., Savenye, W., & Thornburg, H. (2008). *AMEEd Edulink.* Retrieved July 10, 2008. from http://ame4.hc.asu.edu/edulink

Blackboard. (2008). Retrieved July 10, 2008, from http://www.blackboard.com/us/index.bb

Bondendorf, E., Schertler, M., & Cohen, E. (2006) Producing reusable web based multimedia presentations. In Alex Koohang (Eds.), *Interdisciplinary Journal of knowledge and Learning Objects, 2*, 127-142.

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., & Vanderdonckt, J. (2003). A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, *15*(3), 289–308. doi:10.1016/S0953-5438(03)00010-9

Cesarini, M., Monga, M., & Tedesco, R. (2004, March 14-17). Carrying on the e-learning process with a workflow management engine. In *Proceedings of the 2004 ACM Symposium on Applied Computin*g (pp. 940-945), Nicosia, Cyprus. SAC '04. New York: ACM.

Claroline (2008). Retrieved July 10, 2008, from http://www.claroline.net

Costagliola, G., Ferrucci, F., & Fuccella, V. (2006, July 11-14). SCORM run-time environment as a service. In *Proceedings of the 6th international Conference on Web Engineering* (ICWE '06), Palo Alto, CA, USA, (vol. 263, pp. 103-110). New York: ACM.

Cuppens, E., Raymaekers, C., & Coninx, C. (2006). A model-based design process for interactive virtual environments. In S.W. Gilroy & M.D. Harrison (Eds.), *Interactive Systems. (LNCS, 3941*, pp. 225-236.) Springer, Heidelberg.

Dillenbourg, P. (2000). Virtual learning environments. In *Proceedings of Workshop on Virtual Learning Environments during EUN Conf. 2000, Learning in the new millennium: Building new education strategies for schools*.

Docq, F., Lebrun, M., & Smidts, D. (2007). Claroline, une plate-forme d'enseignement/apprentissage sur Internet: pour propulser la pédagogie active et l'innovation? In M. Frenay, B. Raucent, & P. Wouters (Eds.), *Actes du 4ième colloque "Questions de pédagogies dans l'enseignement supérieur* (pp. 99–109). Presses universitaires de Louvain, Louvain-la-Neuve.

Dokeos, (2008). Retrieved July 10, 2008, from http://www.dokeos.com/

Dougiamas, M., & Taylor, P. C. (2003, June 23-28). Moodle: Using learning communities to create an open source course management system. In *Proceedings of the World Conference on Educational Multimedia Hypermedia & Telecommunication (ED-MEDIA'2003)*, Honolulu (pp. 171–178). Charlottesville: AACE Press.

Florins, M., Montero, F., Vanderdonckt, J., & Michotte, B. (2006, May 23-26). Splitting Rules for Graceful Degradation of User Interfaces. In *Proceedings of the 8th International Working Conference on Advanced Visual Interfaces (AVI'2006)* Venezia (pp. 59-66). New York: ACM Press.

Germán Sánchez, V., Téllez Mora, F., & Morales Gamboa, R. (2007). Collaborative environment for the composition of learning scenarios. In Gelbukh & Zechinelli (Eds.), *IEEE Editorial. ENC07 Conference*, Mexico.

González Calleros, J. M., Vanderdonckt, J., & Arteaga, J. M. (2007, June 6-8). A method for developing 3D user interfaces of information systems. In *Proceedings of the 6th International Conference on Computer-Aided Design of User Interfaces (CA-DUI'2006*, *5*(7), pp. 85-100). Springer-Verlag, Berlin. Guerrero, J., Vanderdonckt, J., & Gonzalez, J.M. (2008). FlowiXML: A step towards designing workflow management systems. *Journal of Web Engineering*, *4*(2), 163–182.

Guerrero, J., Vanderdonckt, J., González, J. M., & Winckler, M. (2008). Modeling user interfaces to workflow information systems. Paper presented in *4th International Conference on Autonomic and Autonomous Systems* (ICAS'2008). Springer-Verlag.

Jonassen, D. J., & Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning environments. *Educational Technology Research and Development*, *47*(1), 61–79. doi:10.1007/BF02299477

Li, Q., Lau, R. W., Shih, T. K., & Li, F. W. (2008). Technology supports for distributed and collaborative learning over the internet. ACM Tranactions on Inter. Technology, 8(2).

Limbourg, Q., & Vanderdonckt, J. (2004). UsiXML: A user interface description language supporting multiple levels of independence. In M. Matera & S. Comai (Eds.), *Engineering Advanced Web Applications* (pp. 325-338). Paramus: Rinton Press.

MacDonald, C.J., Stodel, E.J., Thompson, T-L., Muirhead, W., Hinton, C., Carson, B., & Banit, E. (2005). Addressing the eLearning contradiction: A collaborative approach for developing a conceptual framework learning object. Interdisciplinary Journal of knowledge and Learning Objects, 2, 79-98.

Margain, L., Muñoz, J., & Álvarez, F.J. (2008). A *Methodology for Design Collaborative Learning Objects*. I*CALT 2008*, Santander, Spain.

Martínez Ruiz, F. J., González Calleros, J. M., Guerrero García, J., Muñoz Arteaga, J., & Vanderdonckt, J. (2008, June 9-11). ECOOL: Genaration of Collaborative Multiplatform Scenarios with Interactive Learning Objects. In M. Lozano & J.A. Gallaud (Eds.), *Servicio de Publicaciones de la Universidad de Castilla - La Mancha* (pp.361-370), Albacete.

Michotte, B., & Vanderdonckt, J. (2008, March 16-21). GrafiXML, A Multi-Target User Interface Builder based on UsiXML. In *Proceedings of 4th International Conference on Autonomic and Autonomous Systems (ICAS'2008)*, Los Alamitos: IEEE Computer Society Press

Paternò, F. (1999). Model-based design and evaluation of interactive applications. *Applied Computing.* Springer*:* Heidelberg.

Preece, J. (2000). Online communities*:* Designing usability and supporting socialability. John Wiley & Sons, Inc.

Sloodle. (2008). Retrieved May 10, 2008, from http://www.sloodle.org van der Aalst, W., & van Hee, K. (2002): *Workflow Management: Models, Methods, and Systems*. Cambridge: The MIT Press.

Steinkuelhr, C. A., Derry, S. J., Woods, D. K., & Hmelo-silver, C. E. (2002). The STEP environment for distributed problem based learning on the World Wide Web. In *Proceedings of Computer Support for Collaborative Learning (CSCL 2002)*.

Vanderdonckt, J. (2005, June 13-17), A MDA-compliant environment for developing user interfaces of information systems. *Proc. of 17th Conf. on Advanced Information Systems Engineering (CAiSE'05)*, Porto (LNCS, 3520, 16-31). Berlin: Springer-Verlag.

Wainwright, K., Osterman, M., Finnerman, C., & Hill, B. (2007, October 7-10). Traversing the LMS terrain. In *Proceedings of the 35th Annual ACM Conference on User Services SIGUCCS'2007* (pp. 355–359), Orlando. New York: ACM Press

Yingling, M. (2006). Mobile Moodle. *Journal of Computing Sciences in Colleges*, *21*(6), 280–281.