Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

Miguel J. Hornos and María V. Hurtado
Software Engineering Department, University of Granada,
E.T.S. de Ingenierías Informática y de Telecomunicación, Campus de Aynadamar, 18071 Granada, Spain
Email: {mhornos, mhurtado}@ugr.es

M. Pilar Fernández-Sánchez
Department of Quantitative Methods in Economics and Business, University of Granada
Faculty of Business Studies, Campus de Cartuja, 18071 Granada, Spain
Email: pilarfs@ugr.es

Álvaro López-Martínez, Kawtar Benghazi, María L. Rodríguez-Almendros and María M. Abad-Grau
Software Engineering Department, University of Granada,
E.T.S. de Ingenierías Informática y de Telecomunicación, Campus de Aynadamar, 18071 Granada, Spain
Email: arsu@correo.ugr.es; {benghazi, mlra, mabad}@ugr.es

Educational software should be designed by development teams with actual teachers. Thus, implementations of this type of software would be able to cover most teaching requirements and to adjust to different teaching principles. Moreover, some more work should be done to abstract common patterns in educational software so that software engineering methods can be adapted to suit educational needs. We propose an abstraction of system users by defining a role hierarchy which fits web-based collaborative educational systems. This proposal, which can significantly improve software adaptability and usability, has been used to develop Tutor, a web-based educational software system to promote a blended learning environment in Higher Education.

Keywords: Web-based educational systems, hierarchical role-based design, usability, adaptability, blended learning, Higher Education

ACM Classification: D.2.2 Design Tools and Techniques; K.3.1 Computer Uses in Education

1. Introduction

In Higher Education context, the new Information and Communication Technologies (ICT), and especially the Internet technologies, facilitate the access to resources and services, as well as remote communication and collaboration. However, ICT alone cannot significantly improve the quality of learning and teaching, and like any other media, ICT must be carefully integrated into educational activity (Lipponen and Lallimo, 2004; McPherson and Nunes, 2008). Studies of teaching and learning at different levels indicate that successful deployment of ICT tools depends on addressing the pedagogical issues associated with effective quality services offered as teaching and learning support (Taylor, 1996; Chen, Hsu and Hung, 2000). The integrated combination of traditional learning with web-based online approaches is known as blended learning (Singh, 2003;
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

Garrison and Vaughan, 2008; Gerber, Grund and Grote, 2008). This approach considers the teacher as a facilitator who supports students in collaborative learning. In this context, most interactions take place in face-to-face settings with the web acting as a kind of backbone for the transmission and sharing of materials, ideas and experiences, for its preparation and organization, and for administration tasks (Derntl and Motschnig-Pitrik, 2004). ICT therefore offer the adequate means for this kind of interactive learning environment, providing tools to assist teachers in their tasks rather than pillars on which teaching/learning must be based.

Nevertheless, the design of ICT-based educational software has primarily focused on learning and has paid little attention to the teaching side, and as a result, teachers usually find it difficult to integrate it into their classes. Moreover, the teaching staff (as end users) should be involved in the development and implementation process from the very beginning in order to successfully implement educational technology in a pedagogically meaningful way (Ligorio and Veermans, 2005).

Educational software must therefore be developed following the requirements of educational sector experts. It should also assist teachers following different educational theories, known as student-centred approaches, to combine or integrate the traditional learning with web-based online approaches. Although a lot of web-based educational systems claim to draw upon theoretical positions, most of them do not explain how theoretical approaches may be supported by ICT. Generally speaking, very little flexible and adaptive educational software has been developed (Lin, Young, Chan and Chen, 2005).

To develop this kind of software, main teaching/learning requirements must be considered (Kerres and De Witt, 2003) so that traditional modeling can be changed for doing software engineering approaches more suitable to different educational requirements. For these systems to support collaborative and cooperative teaching and learning, we believe that usage-centred design (Constantine and Lockwood, 2002) can be enhanced by using a hierarchy to model relationships between user roles. This new hierarchy can help to improve the flexibility and adaptability (Fayad and Cline, 1996) of the web-based application components developed on it. Based on this schema, user interaction in educational systems is modeled by the relationships between the different roles a user can play.

Our approach has been applied to a web-based educational system for blended-learning support, called Tutor (2012), which is available at http://tutor.ugr.es and has been developed at the University of Granada, Spain. The idea is to use Tutor as a web-based teaching support system in order to assist classroom education with different groups of university students. Tutor is a flexible, intuitive system which is designed to streamline the processes of managing and tracking information relating to different subjects, as required by the European Space for Higher Education (Communiqué of the Conference of Ministers responsible for Higher Education, 2003). Furthermore, Tutor provides interactive electronic services, theory and practice learning materials, and other useful information, as well as new CMC (Computer-Mediated Communications) and CSCA (Computer-Supported Collaborative Argumentation) tools, terms which include both electronic asynchronous communications (e.g. discussion forums and internal messaging between registered users) and electronic synchronous communications (e.g. chat rooms and instant messaging). Staff of different departments teaching subjects on academic degree courses at the mentioned university can currently benefit from this system, since collaboration and cooperation is not only student-focused but also teacher-focused.

The rest of the paper is organized as follows: Section 2 focuses on how current web-based tools used for blended learning in Higher Education fulfil goals of user adaptability and therefore
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

usability. Section 3 introduces the hierarchical design of user roles as a model for collaborative environments, which has been developed to solve the main problems that usage-centred approach poses in web-based systems for Higher Education. Section 4 shows how this new design has been applied to Tutor, a web-based educational platform specially intended for blended learning in university courses, as well as the user roles designed for it and the most important services offered by each of them. It also explains the main technical features of the system related to user roles. Section 5 briefly presents the results of the usability evaluation we have conducted to assess how Tutor has been valued and accepted by teachers and students. The final section outlines the concluding remarks and future lines of research.

2. Web-based Tools for Blended-learning Support in Higher Education:
   Usability and Adaptability Requirements

Perhaps the most important change in Higher Education that is going on is the whole concept of the educational process. This huge change can be observed just by considering the language evolution. Words such as ‘teaching’, ‘hand notes’, ‘memorizing’ or ‘lecture’ are being replaced by ‘learning’, ‘document (web) search’, ‘discovering’ and ‘discussion’ respectively. Thus, students must be in the core of the educational process playing an active role. These changes are even being regulated in very broad frames, as occurs with European universities and their European Space for Higher Education, a high level institution pursuing these changes among one of its most important goals.

For software to be really useful, analysis and hence development must be customer-driven, i.e. considering those in charge of the company service or department requiring the software. The software must therefore perform tasks so that customer requirements are fulfilled. In our context, we can consider teachers and coordinators in charge of a subject as direct customers, and their students as indirect customers. Because teaching practice can differ greatly according to the teacher, the software must be flexible and adaptable to allow different blended teaching methods in order to create the appropriate learning spaces for the students.

Together with adaptability criteria, usability must be also crucial in this type of software, as it is the main feature for a system to be accepted and used. To guarantee adaptability and usability requirements, we also have to apply quality hypermedia design principles (Arroyo, Medina, Hornos and Molina, 2007).

Although there are currently hundreds of web-based educational environments, most of them lack flexibility and usability and, in terms of educational software, this results in rigid tools which are unable to suit the requirements of teachers following different educational methods.

Certain projects have been created with a strong basis on pedagogical models, such as ITCOLE (Innovative Technologies for COllaborative LEarning) (Ligorio and Veermans, 2005; Rubens, Emans, Leinonen, Skarmeta and Simons, 2005). In this project, modular knowledge-building environments were developed, based on pedagogical models of collaborative learning. LauLima system (Breslin, Nicol, Grierson, Wodehouse, Juster and Ion, 2007) comprises a shared workspace and a digital repository for the creation of dynamic wiki pages. This system allows student teams to map their design process from the beginning to the end and offers great flexibility, since students can access and manage resources online at any time from any location, and they can collaboratively manage their learning and workflow. Chou, Wu, Li and Chen (2009) present a web usage mining approach based on the sequence mining technique applied to e-learner’s navigation
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

behaviour. This approach allows discovering patterns in the navigation of e-learning websites for personalization in e-learning environments. Another interesting project on web-based educational tools is “School for all” (Lin et al, 2005), which integrates different teaching methods and pedagogical models in order to satisfy the different needs of instructors. “School for all” can be considered a milestone for a new approach in educational software development. This approach, called adaptive teaching in Lin et al (2005), consists of a web-based system in which goals and user profiles can be decided by the users. Consequently, this type of system can be considered as a tool for translating educational principles into educational practices. It also enables teaching-learning processes to be more easily incorporated into the educational model. Besides, this approach facilitates the integration of constructivist and objectivist pedagogical models (Brown and Campione, 1996; Wolfe and McMullen, 1996).

However, do they really fulfil usability and adaptability requirements? No clear assessment has been published about the conceptual representation of user profiles in the web-based tools referred above. While many of them adopted the classic user-centred approach, some of them seem to use a usage-centred engineering (Constantine and Lockwood, 2002). As an example, “School for all”, in which the same person can have different roles depending on the subject, and this is called adaptive co-teaching. A privilege-setting module allows an instructor to add a teaching team and set roles as co-teacher, consultant or assistant. Privileges are also given by the subject coordinator for each teacher connected with the subject, and so syllabus settings, bulletin boards, discussion forums, evaluation and material management can be accessed depending on the privileges granted. Most of these features are included in our proposal, which is explained in next section, but with the difference that we make explicit the conceptual representation of the roles that our system uses as well as the relationships between them in a hierarchical model which enables applying inheritance.


In order to fulfil the development requirements mentioned above, we should opt for a design focusing on usage rather than users, i.e. a usage-centred engineering (Constantine and Lockwood, 2002). In this approach, roles substitute users so that the user is not who accesses the services but the role s/he is playing. Consequently, the classic use case diagram (Bruegge and Dutoit, 2004) for modeling how users interact with the system is mainly replaced by one indicating how roles interact with the system. As an example in web-based educational systems, an instructor would be able to interact with the system by playing different roles: subject coordinator, teaching assistant, co-teacher, etc., and usually a same instructor can play different roles, depending on the subjects s/he teaches. Roles therefore constitute an abstraction of users accessing the system. In fact, each role defines the set of tasks and operations that a user can play (on a given subject, in this type of system). The main advantages of systems under this role-based abstraction are that both adaptability and usability are improved. As a result, this approach facilitates and speeds up the web-based software development.

Other educational software applications, such as “School For All” (Lin et al, 2005), Sakai (2012) and Moodle (Büchner, 2011), are built under the assumption that privileges depend on roles instead of users, and their use is therefore motivated by the fact that roles are not independent of each other, i.e. the sets of tasks defined by the different roles are not unconnected. Furthermore, privileges are not directly related to users but to roles, and so there should be a one-to-one relationship between roles and privileges. By way of example, an instructor under the role of
teaching assistant for a given subject should have certain privileges because of his/her role, e.g. the teacher could access discussion forums but not grant access to other users, or s/he could upload material for students but not delete already uploaded material, etc. Versions previous to Moodle 1.7 only allowed static and predefined user roles. Since Moodle 1.7 the creation of new roles may be dynamically performed.

However, usage-centred engineering does not consider relationships between roles, i.e. roles are used as mutually independent. Roles sharing tasks cannot therefore benefit from the characteristic inheritance mechanism of the hierarchical models used in object-oriented software development. As a result, adaptability and therefore usability cannot be assured at its most extreme. Thus, the usage-centred approach should be enhanced in order to better model the different roles that users can play in a web-based system for blended learning. The closer the model to the reality, the better our goals of adaptability and usability will be achieved.

To solve this issue, we have defined a new design feature, called HRD (Hierarchical Role-based Design), which is able to represent relationships between roles and allows improving user adaptability and usability as a consequence in educational software. While HRD can be applied to many different web-based applications, the educational environment can take great advantage of this approach. HRD is the result of using object-oriented design (Bruegge and Dutoit, 2004) to define roles and the relationships between them right from the beginning of the development process, i.e. in the use case diagrams. Object-oriented programming classifies operations as belonging to different types of data (called classes) so that only objects of a given class can perform the operations defined by that class. The main feature that allows the object-oriented approach to improve adaptability is the use of inheritance. All the objects belonging to a subcategory (called a subclass) of other class (called the superclass) can inherit operations and other characteristics from it. They can therefore perform the same operations as those in the superclass, and even refine and adapt some of these operations (i.e. the subclass is a specialization of the superclass). At the use case level, the object-oriented approach classifies users as belonging to different categories. As HRD is usage-centred, roles instead of users are categorized and the hierarchy is therefore defined not between classes, as it occurs at a further stage, nor between users, as it occurs in user-centred approaches, but between user roles. Our proposal, which is based on this approach, allows taking advantage of the object-oriented design benefits at a more abstract level. Consequently, user roles and indirectly high level tasks, i.e. use cases, are hierarchically defined and thus HRD allows a quick development of more adaptable systems. As a result, by using HRD to represent the hierarchical model of user roles of a system, software adaptability (Fayad and Cline, 1996) can be achieved with minimal costs. Therefore, a new system functionality pursuing improved usability may inherit some tasks from one already launched, and a new role may also inherit associated tasks of another already defined. The final purpose is to have a flexible system able to suit different educational models the teachers may need, by creating and/or adapting the roles needed for the corresponding educational model.

Moreover, a system applying HRD would easily benefit from using Lightweight Directory Access Protocol (LDAP) servers. LDAP is a standard protocol encouraged for multi-platforms web-based applications with read-intensive operations. Many times a person under a student role accesses an educational system just to collect some information or documents. The hierarchical role design will have a direct counterpart in a directory tree used by LDAP.

In the next section, we will apply HRD to the user roles used in a real web-based educational system, and we will see how roles can be easily and rapidly created for students and teachers and
how services will be assigned to them. Moreover, we will be able to adjust the services according to the needs of a particular role.

4. Tutor as a Case Study

This section presents the application of HRD to the roles used in the web-based educational system called Tutor (2012), and explains the roles considered and the most important services each one offers, as well as the main technical features of Tutor. This web-based educational platform, which constitutes the main result of three successive teaching innovation projects we have carried out, received an honourable mention in Teaching Innovation Awards granted by the University of Granada in 2008. The central objective of these projects was to propose a set of tools or utilities intended to build web-based systems for supporting teaching and learning processes, and for the management of academic data related to university subjects. Tutor is currently being used by many teachers and students of the University of Granada. More concretely, there are more than 8000 students and 84 teachers registered on it, as well as 90 subjects belonging to 25 different degree courses and taught in 10 schools and faculties of such university are managed by our system at the moment of writing this paper.

As a web system, Tutor is permanently available so that students can access the system whenever they want and from anywhere with an Internet connection. Although the system is especially intended to support teaching and learning in a blended learning framework, it is also appropriate for students who cannot regularly attend classes, since it gives them the opportunity to “follow” the corresponding subjects more successfully.

4.1. Role-centred Services Provided by Tutor

Each tool, functionality or service provided by Tutor is role-centred. Therefore, whenever users access the system, they are offered services depending on the roles they are allowed to play. All the services can be considered useful for anyone regardless of his/her preferred didactic method. For reasons of conciseness, from here on we shall focus directly on ICT-based tools instead of on the theoretical approaches under them in order to describe the services provided by Tutor.

Figure 1 shows the hierarchical model designed for the user roles used in Tutor. For each subject, a user accessing the system will be able to use all the services allowed by the role s/he plays for that subject.

The most restricted user role is that of casual user, which is played by every Internet user accessing the system without authentication. In addition to basic information services, such as a general description and diverse information about Tutor, casual users can
access public information about the teachers and subjects registered in the platform (e.g. class and tutoring timetables, subject syllabus, bibliography, evaluation system, etc.). They can also contact the system administrator or any registered teacher to make suggestions or ask any question about the platform or any subject managed on it. These services are accessed through the last four upper circular buttons (see Figure 2).

For authenticated users, the system distinguishes between several roles. While there are two roles for students (i.e. course student and lab student), there are three roles for educators: lab teacher, course teacher and course coordinator (see Figure 1). Each user can play a unique role for each subject with which s/he is related in the system, but the role s/he plays for a subject can be different to the role s/he plays for another one. These roles are assigned to each student at the moment of enrolling on a theoretical and/or practical group of a given subject. Thus, if the role played by a student for two given subjects are course student and lab student respectively, this means that s/he is only registered on a theoretical group for the former, whereas s/he is registered on both a theoretical and a lab group for the latter, being the former more restricted (it allows accessing less services or functionalities) than the latter. Likewise, teachers are assigned the corresponding roles when they create one or more theoretical and/or practical groups for a subject in the platform. Only one teacher per subject may play the course coordinator role, which is decided by all the colleagues teaching that subject. As an example, a teacher at a certain institution could be a course teacher of one subject (i.e. s/he explains the theoretical part of the subject), the course coordinator of another subject (with total access to all teacher functionalities for that subject), and only a lab teacher for a third subject (this being the most restricted role for teachers). Consequently, a same
<table>
<thead>
<tr>
<th>User role</th>
<th>CMC/CSCA services</th>
<th>Other services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casual user</td>
<td>• Contact form for sending any question or suggestion related to the platform and/or the subjects registered on it.</td>
<td>• Public information about the platform itself, and about each teacher and subject registered on it, such as class and tutoring timetables, syllabus, bibliography, evaluation system, etc.</td>
</tr>
<tr>
<td>Course student</td>
<td>• Using the discussion forums, internal messaging and chat, and reading the notices placed by his/her teachers and the platform administrator in the notice board. • Accessing virtual tutoring sessions and complementary virtual classes.</td>
<td>• Registering on a lab group, consulting exam marks, filling out and updating the electronic student card, accessing didactic materials (such as slides, exercises, solutions, etc.), completing exams or surveys, uploading files containing the work carried out for a specific activity (proposed by a teacher), etc.</td>
</tr>
<tr>
<td>Lab student</td>
<td>• Accessing complementary virtual laboratory classes and reading notices for a lab group.</td>
<td>• Using functions similar to those used by course students, but restricted to lab groups, such as completing lab exams or consulting lab test marks. • Registering his/her attendance to lab sessions and seeing his/her (attendance) record.</td>
</tr>
<tr>
<td>Lab teacher</td>
<td>• Using the communication services (i.e. notice board, discussion forums, internal messaging and chat), but restricted to lab groups.</td>
<td>• Using operations restricted to lab groups, such as creating his/her groups in the system, entering and updating all the information relating to each group (e.g. timetables, classrooms, etc.), preparing online exams and/or self-evaluation tests, registering exam marks, etc. • Managing and consulting student data, such as listing the students registered on a group, or changing the password to a student. • Configuring and monitoring the attendance control system for practical classes.</td>
</tr>
<tr>
<td>Course teacher</td>
<td>• Using the communication services for communicating anything to any of the groups of his/her subjects.</td>
<td>• Using operations similar to those used by lab teachers, but extended to theory groups.</td>
</tr>
<tr>
<td>Course coordinator</td>
<td>• Opening discussion forums for all or some teachers and/or students of the subject being coordinated.</td>
<td>• Assigning teachers registered on the platform to the subject, and editing its • Deciding which didactic materials are shared or not among the groups of the subject.</td>
</tr>
</tbody>
</table>

Table 1: Summary of available services in Tutor for each user role
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

authenticated user (student or teacher) can play a different role with respect to a given subject in different academic years. Notice also that although each user can play different roles in the system, s/he can only play one role at a given moment, and this will depend on the subject with which s/he operates at that moment.

Table 1 summarizes the available services according to the role played by the user. For each user role on the left column, only the services that are not available for an ancestor (following the inheritance structure shown in Figure 1) are included, distinguishing CMC/CSCA services (second column) from the rest of services (last column).

The final aim of CMC/CSCA services is to offer new communication channels between teachers and students in order to complement the usual ones used in traditional education. Although learner characteristics, such as gender, sociocultural background and ability, as well as whether or not students have computer skills and attitudes, comprehensive reading scores and popularity with classmates, influence the participation of students (Prinsen, Volman and Terwel, 2007), most students find these new technologies less intimidating than participating in traditional classes when they interact with classmates and join in discussions. These technologies also facilitate critical thinking, even in cultures with certain restrictions in this sense (Chiu, 2009).

Teachers can propose collaborative tasks to their students based on CMC/CSCA services. By taking advantage of these services, teachers can offer their students virtual tutoring sessions (which could be previously scheduled) and complementary virtual classes. The idea is to improve social interaction and create a sense of community (McInerney and Roberts, 2004) among users of our system, something which is very important in collaborative learning.
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

Figure 2 shows the services available for students in Tutor, while Figure 3 displays the main functionalities that teachers can use. One example of a very successful Tutor service for both teachers and students is the system to control attendance to practical classes, which allows students to register their own attendance to a given lab session and allows teachers to annotate comments for each of their students (see Figure 4). Making use of this tool and others included in Tutor, teachers can provide a more effective feedback to their students than using traditional methods (Denton, Madden, Roberts and Rowe, 2008).

4.2. Main Technical Features of Tutor

Figure 5 shows the graphical representation of the system layer architecture. The storage layer is the base of this architecture and consists of a database management system (DBMS) and a file system, which respectively store data and files of the system. At the middle layer is the system core, which consists of common system classes (Users, Subjects, Departments, Faculties and Schools, … ) and basic system functionalities (Controllers, Template engine and Language manager, which will not be explained here, since they are out of the scope of this paper). The highest layer is made up of the modules that implement the corresponding functionalities or services. Each module is developed making use of the well-known Model-View-Controller (MVC) architectural pattern (Reenskaug, 2003). Persistent data of the model resides in the database, which can be accessed through functions and classes. The view and controller parts are implemented in HTML (W3C, 2011b), CSS (W3C, 2011a) and JS (Mozilla Developer Network, 2011).

Services are therefore offered through dynamic (rather than static) web pages which are automatically generated from the contents of the relational database that constitutes the basis of our information system. For this purpose, Tutor has been implemented using mainly PHP5 (2011) with embedded queries to our MySQL5 (2011) database. Consequently, a good design of the database is essential in order to provide these services. In the current implementation, more than 100 tables comprise our database, which stores and relates information about students, teachers,
subjects, groups, degree courses, faculties, departments, etc. The database management system enables all the information to be maintained in a centralized way.

4.2.1. System Modules

As Tutor is implemented on a modular architecture, the functionalities or services that it provides are grouped and implemented into the corresponding module. This allows developing different functionalities in an independent way (even by different developers), without having conflicts between them. Figure 6 shows the 26 Tutor modules which have been implemented so far. In general, these modules have self-descriptive names. Thus, for example, the Downloads module implements the Downloads Area functionality, and the Notices module implements the functionalities related to the notice board.

Statistics module offers several summaries about the system usage to system administrator, such as concurrent accesses, number of students, assignments and files per subjects and academic years, etc.

![Diagram of system architecture](https://via.placeholder.com/150)

Figure 5: Graphical representation of the system layer architecture

![Complete map of current Tutor modules](https://via.placeholder.com/150)

Figure 6: Complete map of current Tutor modules
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

As the Group Manager module implements the services that allows teachers to create the different types of groups, as well as to enrol students in them, this module is in charge of assigning the corresponding role to each user with respect to a given subject and academic year. The rest of the modules will allow a user to use a certain service included in it depending on the role s/he is playing for the specific subject (and academic year) with which s/he is working at that moment.

4.2.2. User Role Management

A new functionality related to HRD has been designed to be implemented and applied to Tutor, as shown in Figure 7, which displays a prototype of a control panel that will allow the system administrator to create the needed user roles and to link the corresponding services to them. A role can be created from scratch or from an already existing role, by using the button Create a new role from this (which is placed to the right of the drop-down list containing the roles already created). In this latter case, the new role will inherit all the services assigned to the (existing) base role, which will be its ancestor in our user role hierarchical model (see Figure 1 by way of example). Moreover, the system administrator will be able to add any other service from the left-side list (which includes all the services or functionalities implemented in Tutor) to the right-side list containing the services assigned to the role being edited, as well as delete some of the services assigned. This tool also enables adapting or refining the services assigned to a given role using the Service properties... button, which allows restricting the corresponding functionalities through a series of options that can be selected in each case. As an example, the Grades service can be restricted for a given role to be applied only to practical groups.

Figure 7: Administrator tool for creating and adapting user roles
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

Figure 8 shows an Entity-Relationship diagram with the new entity ROLE and the relationships required to connect it with already existing entities. As a result, the current database will have 2 more tables, one for the new entity and the other, USER_ROLE, to represent the relationship between USER and ROLE.

Table 2 shows an example of an instance method related to the user role management. The main steps are described by using pseudocode. As an example, code (MySQL query) required in one of the steps to be embedded in PHP is shown. The query assumes the database has been updated according with the Entity-Relation Diagram shown in Figure 8.

**Method:** getListOfGrantedServices
**Class:** User
**Input data:** –
**Output data:** a two-column array with service IDs and service names granted to a user

**Description in pseudo-code:**
1. roles=findDirectRoles()
2. create temporal table ANCESTOR
3. foreach role in roles
   3.1. ancestors=role->findAncestors()
   3.2. Add ancestors to table ANCESTOR
4. services=getListOfServices(user_id)
5. remove table ANCESTOR
6. remove arrays roles, ancestors
7. return services

**MYSQL code for step 4:**
SELECT SERVICE.names, SERVICE.id
FROM USER,USER_ROLE,ROLE,SERVICE,ANCESTOR(USER.role_id)
WHERE USER.user_id=ANCESTOR.user_id
AND USER.user_id=USER_ROLE.role_id

**Table 2:** An example of a method related to the user role management. At the bottom, an example (coded as a MySQL query) required in one of the steps to be embedded in PHP is shown.
5. Evaluation

If ICT-based tools are difficult to understand and use, users will waste more time in learning and training, and thus will generate an unattractive perspective for these tools. However, ICT can be viewed as a way of procuring advantages in the educational context. With the aim of assessing Tutor by its users (teachers and students), a double evaluation study was conducted to analyse both usability and adaptability grade of Tutor tools to achieve their objectives. Details of both studies, which were conducted in May 2010 at the University of Granada, are given in next subsections.

5.1. Interview to Teachers

A semi-structured interview was conducted among teachers using Tutor. 28 teachers voluntarily participated by answering 15 open questions about Tutor and how they can benefit from its tools and services. As the most useful and valued functionalities, interviewees pointed out the tools or services to: control student attendance to practical classes; share didactic material through the downloads area with flexibility, by being able to apply different permissions to each file uploaded; propose tasks to students so that they can upload the files with the result of their work; register exam marks which are shown to students preserving the data privacy, but providing contextual information on how good the student’s result is with respect to the rest of his/her mates; and create and manage different types of groups, not only theoretical and practical ones, but also others for diverse purposes (such as taking an exam, creating small teams to prepare a work collaboratively, visiting a company, etc.). Although the usability, flexibility and adaptability of Tutor tools were in general well valued by the interviewees, we collected some suggestions to create new services and to adapt the existing tools to new applications and practices, as well as to improve their usability in certain aspects. Most of these suggestions were implemented and included in the platform.

5.2. Survey to Students

We would like to emphasize that this is not a traditional survey in the sense of using a sample size in order to generalize the whole population. A pilot test was conducted with the objective of knowing what students using Tutor think of its tools. It was designed to analyse the use that students make of Tutor tools, and whether this platform is useful for them in their academic purposes, as well as to know whether they prefer to carry out certain educational activities using Tutor or in a traditional way. Additionally, we obtained user’s profile (e.g. age, gender, degree course, number of years at university, average score, etc.) as well as information about technology and web uses (e.g. technological devices they had, frequency of computer and Internet use, etc.), because these data can be used to provide a baseline for comparison and contrast between groups of students. This study would help developers and teachers better understand the impact of the platform in their educational context. All the participants voluntarily completed an on-line questionnaire in class (at the computer lab) with 39 questions divided into four sections: one general, to establish the user profile, and three about different aspects of Tutor. Anonymity and confidentiality were stressed.

5.2.1. Data Gathering and Analysis

The survey was completed by 307 students aged 18-38 years ($\bar{x} = 21.9$, $s = 2.64$). This sample was considered large enough to confirm measures and tests. The students were enrolled on four
degree courses: Economics (4 years), Management (4 years), Business (3 years), and Marketing (2 years).

The analysis of the questionnaire answers was carried out using the statistical package SPSS, version 12. As a preference survey, most of the questions result in dichotomous response, i.e. responses that admit two possible alternatives, such as yes or no. In addition, groups of respondents attending to two population characteristics, gender and age, were established in order to determine whether there were significant differences between these subsamples. The new groups are independent random samples from binomial distribution, and they are large enough so that the sample distribution of the difference of proportions can be adequately approximated by a normal distribution. Consequently, an independent sample test and a non-parametric test (Mann-Whitney test) were used to look for the differences, depending on whether or not the answer to the question had more than two options. Statistical significance was set at 0.05 for all the statistical analysis.

5.2.2 Results and Students’ Opinion

For reasons of conciseness, only the main results have been included in this subsection. Related to user profiles, 56.7% were female and 43.3% were male, while students under the age of 21 represented 44% of the whole population, participants aged 21 to 25 were 48.3%, and students over 25 corresponded to only 7.8%. These groups were used to analyze whether there were significant differences in responses according to age. Most students (66.95%) were in their first year (35.7%) or their fourth year (31.2%), 16.2% were second-year students and 14.3% were third-year students.

Regarding the use of ICT, we can conclude that students participating in this study were ICT users (92.5% of them had a computer, 82.5% had a printer, 75.2% had a mobile phone and 46.6% had a scanner), and most of them (85.9%) used the Internet frequently, i.e. either daily (53.3%) or several times a week (32.6%).

To analyse students’ opinion about different tools or services offered by Tutor, these were classified according to their functionality (into communication tools and other management tools/services) or accessibility (authenticated and free access).

Percentages for use of the different tools analysed in the pilot study are shown in Figure 9, which indicates that authenticated access services were used more than free access ones. The majority of students have used the management tools (lab utilities, downloads area, grades query, teachers information, syllabus/bibliography, etc.). However, the percentage of students using communication channels was particularly low (16.9%). This is all the more surprising if we take into account that 92.5% of students consider these services to be useful. Looking for a reason for the difference between use and usefulness, we studied the responses of people who said they had not used communication channels. We found that 57% were female and 43% male. The majority of the group was between 21 and 25 years old, 93% had a computer, and 82% had access to Internet. We therefore believe that one possible reason might be that students had not been sufficiently motivated by their teachers. Hence, teaching/learning methods intended to foster participative discussion and collaboration among mates, such as virtual debates, should be encouraged in Higher Education context (Benghazi, Hurtado, Bermúdez-Edo and Noguera, 2010).

We also used this pilot study to discover whether students preferred to carry out certain academic activities using Tutor or in the traditional way. The percentages of the students who preferred to
use Tutor for these actions can be found in Figure 10. In most cases, the traditional method implies, for example: obtaining information directly from teachers or peers, copying down didactic materials, or looking for exam grades on a physical notice board at the faculty. We expected Tutor to make life easier for students, and this seems to be the case when we look at the answers (only for tutoring sessions they considered the traditional method to be more comfortable and adequate).
In the answers relating to usefulness, there was only one difference (reaching a significant level at p-value = 0.008) and that was for the usefulness of communication channels, with males achieving lower scores (0.88) than females (0.96). Surprisingly, under-21 students (who are supposed to be more skilled at ICT tools) consider less useful to use communication channels than other students. Both groups have similar scores, with only 11% difference between them, but this constitutes a statistical difference (p-value = 0.003). Additionally, male students obtained significantly higher scores than female students for downloading didactic materials (p-value < 0.002).

6. Conclusions and Future Work

The rapid growth in the development of web-based educational tools has not been always accompanied by meticulous planning and analysis of system requirements. However, we think these are the most crucial stages of the software life-cycle in which the customers (i.e. educational professionals) must intervene for the software to be successful. To elaborate our proposal and develop the system based on it, a group of university teachers has been involved from the very beginning, but this is not a common practice in other developments. As there is much to be done to adapt software engineering models and techniques to the field of education, we encourage the integration of educators within the development team to get more flexible and usable educational software.

One feature of these systems is the existence of different roles that each user (especially educators, but also students) can play depending on the subject. We have introduced a model, called HRD, to represent the hierarchical structure existing among these roles and establish relationships between them, as a way of improving adaptability and therefore usability of educational software for collaborative environments. Among the main advantages of this model, we highlight that it is easy to expand the system with new roles (from the existing ones) to meet the diverse requirements of teachers carrying out different educational practices by using the inheritance mechanism between roles defined by HRD.

We have integrated this model into the development of Tutor, a web-based educational system intended as a flexible and adaptable tool for teaching and learning, and useful for blended learning practices. To improve user adaptability and therefore usability, we have proposed a new administrator tool for the easy management of user roles, which will allow creating new roles (from other existing ones or from scratch) and assigning them the required services, as well as adjusting the functionality provided by a given service assigned to a role by choosing the adequate options within the properties defined for that service.

By conducting a pilot test on Tutor we have investigated the end-user acceptance of our system and identified ease of use and usefulness as two of its main advantages. Like any other software information system, user acceptance and usage are aspects of primary importance for measuring the success of the system. We are working on the design of a more elaborate survey based on the Unified Theory of Acceptance and Use of Technology (Venkatesh, Morris, Davis and Davis, 2003) to analyse more deeply the assessment that students make on both our platform and teachers’ attitude to it, as well as to correlate these data with the use that they make of different Web 2.0 tools. We also plan to elaborate a survey to analyse how well Tutor fits to the teaching principles and methodologies applied by the different teachers using Tutor.

In the future, we plan to extend Tutor by means of a Content Management System (Mauthe and Thomas, 2004). Using the advantages of this technology to expand teaching and learning opportunities is particularly crucial, since they allow users not only to publish contents faster, but
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

also to improve basic tasks such as deciding what to publish and where, collaboratively creating the contents for the web, and publishing them. Hence, we find this technology effective in interactively promoting knowledge sharing between online students. Moreover, and in order to gain interoperability with other educational systems and reusability, we will consider blended learning standards.

References


Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education


Biographical Notes

**Miguel J. Hornos** is a lecturer at the University of Granada (Spain) since 1991, where he received a PhD in computing. He is currently involved in various (European, national, regional and local) R&D&I projects and is serving as co-chair and Program Committee member of several international workshops and conferences. His fields of research are specification and verification of complex software systems, software architectures for developing collaborative applications, ubiquitous computing and intelligent environments, and ICTs applied to business and education, on which he has written several textbooks as well as around 80 research papers published in international journals and conferences specialized on these topics.

**María V. Hurtado** is an associate professor of computer science at the University of Granada, Spain. She got her PhD from the same University in 2002. Her research interests have been focused on enterprise knowledge and information systems modeling, business process modeling, ontology-driven approaches to systems and software engineering, software evolution and learning technologies. She has published more than 70 papers in international conferences and journals. She has been a member of program committees and referees of different international conferences, and she is also part of the editorial board of the Information & Management journal. Currently, she is also involved in several national and international projects.
Hierarchical Role-based Design of Web-based Educational Systems for Blended Learning in Higher Education

M. Pilar Fernández-Sánchez is MSc in economics (1996) from the University of Santiago de Compostela (Spain). She started as an assistant professor in business statistics at the University of Granada in 1997 where she received her PhD in Quantitative Methods Applied to Social Science (2002). She has published several research papers related to risk management or insurance and participated in different projects related to these issues. She has also been Vice-Dean of the Faculty of Economics in Granada in 2004. Now her aim is focused on operational risk although she participates in interdisciplinary projects such as computers, education, social issues.

Álvaro López-Martínez has a Bachelor of Science Degree in computer engineering. His career is focused in web development and systems administration, being the main programmer of the platform mentioned in the article. He has been involved in several projects and papers related to the web engineering field. Currently he is working in the private sector as software analyst and programmer for web applications.

Kawtar Benghazi is currently an assistant professor of computer science at the University of Granada at Ceuta, Spain. She received her PhD from the same university in 2009. Her research interests include formal specification and verification of real-time systems, software oriented architectures and business process modeling. She has published more than 60 papers in international conferences, workshops and journals and she has co-organized several conferences. At present, she is serving as a program committee member and reviewer of different international conferences and journals. She recently co-edited a special issue in Science of Computer Programming journal.

María L. Rodríguez-Almendros received a BS degree in computer science engineering from the University of Granada, Spain, in 1990, and the PhD degree from the same university, in 2002. Her areas of interest include groupware interfaces, usability engineering, notations for specifications and modeling in particular applied to interactive and cooperative systems, and the application of ICTs to people with special needs and the elderly, specifically in the context of educational environments, e-health and ambient assisted living. She has been a researcher on several projects related to collaborative learning and alternative-augmentative communication systems for elderly people.

María M. Abad-Grau received her PhD in computer science from the University of Murcia, Spain, in 2002. She is associate professor at the Department of Computer Languages and Systems, University of Granada and her main research interest is the application of machine learning to discover genetic bases of complex diseases.