Adapting the Telecommunication Engineering Curriculum to the EEES: a project based learning tied to several subjects

J.F. Díez-Higuera, M. Antón-Rodríguez, F.J. Díaz-Pernas, M. Martínez-Zarzuela, D. González-Ortega, D. Boto-Giralda, M. López-Coronado, B. Sainz-de Abajo, I. de la Torre-Díez
Department of Signal Theory, Communications and Telematics Engineering
Telecommunications Engineering School, University of Valladolid
Valladolid, Spain
{josdie, mirant, pacper, marmar, davgon, danbot, miglop, beasai, isator}@tel.uva.es

Abstract—**This paper describes the adaptation process to the European Credit Transfer System requirements of several subjects aiming at the Information and Communication Technologies (ICT) learning. Specifically, these subjects are sited at the Telecommunications Engineering studies lectured in the University of Valladolid. In a first step two first grade subjects have been established, while in a second and final step, coinciding with the new degrees beginning, it will be extended to five subjects placed in consecutive semesters.**

The global programming has been divided into several subprojects of growing complexity, developed into subjects sited in different and successive semesters of the degree, following a pathway leading to the development of a global project throughout four years. The whole learning process is ICT-supported, as tools for overcoming distance and scheduling barriers are offered. In particular, Moodle platform is used, which has been enhanced with self-evaluation and co-evaluation tools developed by the teaching group. Main innovation regarding to the classical approach consists of a computer programming subject focused on the student learning and based on the detailed specification of the activity the students have to perform in and out of the classroom in order to achieve the educational objectives of each of the subjects. The educational strategies used to accomplish these objectives are based on the cooperative learning, on the teamwork developing a programming project (Project Based Learning, PBL), and on the discovery learning.

Keywords: European Credit Transfer System; engineering subjects; Project Based Learning, Moodle platform.

I. INTRODUCTION

In the 50s, 60s, and 70s of the 20th century the idea about learning as a process of gaining knowledge is established. This process takes place when the student can permanently store new information in memory. In this model, the role of the student is to gain knowledge in a passive way, and the teacher's is to design environments where the student can receive a great deal of information [1].

As of 80s and 90s, the teaching idea is transformed into a concept of building up knowledge, which is performed when students directly participate in that building connecting new learning with the foreknowledge [4]. The teacher's task in this case is to design environments where important interactions among the students, the teacher, and the scholarly resources can be provided in order to let the student select, organize, and properly apply the new information.

The problem based learning, and its variant Project Based Learning (PBL), is a strategy which presents the students the challenge of "learning to learn" through the collaborative resolution of open problems, being guided by an enabler. That is, this methodology is based on a project or plan development following the project design approach. Activities are oriented to plan a solution of a complex problem. Work is performed in teams. The students have more autonomy than in a traditional class, they use different resources as the teacher is not the main source of information, he/she acts as an enabler providing resources and offering advice while they advance in their research. The students gather and analyze the information, make discoveries and report the obtained results. The teaching and facilitation are oriented by a wide range of explicit learning objectives; some of them can be precisely focused to the specific content of the theme. The students can achieve additional goals (no planned) as they explore complex themes along different lines; they learn to learn between them; they learn how to help their partners learning; they learn to peer-review and give constructive feedback both for them and their partners. Hence, the aim of the project based learning is to place the student in a realistic scenario, facing real-world issues, to initiate his/her learning process [6]. The PBL has been used in the university in general, and in the area of engineering in particular [5]. The project is the point which unleashes the learning process, and its development causes the knowledge structuring into modules grouping several traditional subjects. Therefore, this working practice depicts a consistent alternative with the teaching method needed for training professionals, especially technicians. This approach is also suitable, to a greater or lesser degree, in mostly every field of study.

978-1-4244-6571-2/10/$26.00 ©2010 IEEE April 14-16, 2010, Madrid, SPAIN
IEEE EDUCON Education Engineering 2010 – The Future of Global Learning Engineering Education

II. BACKGROUNDS

In the 2005/06 school year we launched the adaptation process to the European credit system requirements in one of the subjects design for the Computer Programming learning – "*Fundamentos de Programación*"- undertaken at the Telecommunications and Electronic Engineering studies lectured in the University of Valladolid. Main innovation regarding to the classical approach consists of a computer programming subject focused on the student learning and based on the detailed specification of the activity the students have to perform in and out of the classroom in order to achieve the educational objectives of the subject. The educational strategies used to accomplish these objectives are based on the cooperative learning, formally structured by the puzzle technique, and on a teamwork development of a programming project of low-medium complexity level. This approach has been the result of the learning program adaptation of the Introduction to Computers subject –"*Introducción a los Ordenadores*"- undertaken at the Castelldefels Polytechnic School, a good standing center for its high quality teaching, for its wide experience in the subject adaptation to the European credit system requirements, and for being one of the pioneers in applying the cooperative project-based learning in Spain.

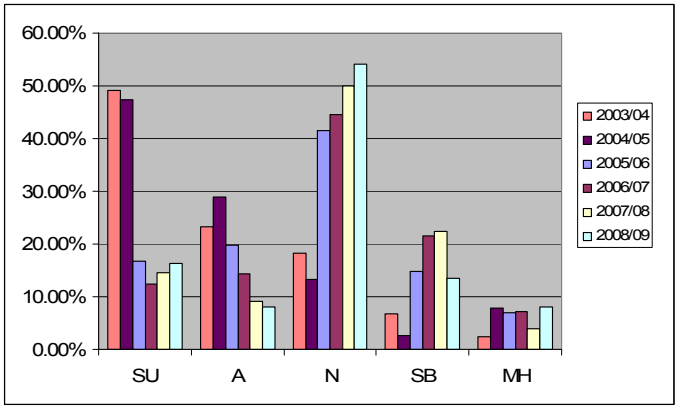

Figure 1. Data trend of the Computer Programming subject grades.

Results of this adaptation are shown in Figure 1. It can be observed failure rates ("SU") drop sharply since 2005/06, while pass rates ("A") decrease at the expense of higher rates ("N", "SB", and "MH"). Therefore, it can be inferred, using this student-centered new methodology, more students pass the subject and with better marks.

According to the promising results obtained in this trial experience over a single subject, last year this approach was extended to a new subject in the second semester, Programming of Multimedia Applications –"*Programación de Aplicaciones Multimedia*"-, thus incorporating a new element to this approach: a multidisciplinary project along a learning thread. Thereby, the project performed during the first semester in the Computer Programming subject is broaden, modified, and completed in the second semester during the Programming of Multimedia Applications subject.

III. MODEL-VIEW-CONTROLLER PROGRAMMING PARADIGM

After considering several organizational schemes, the model-view-controller (MVC) paradigm was considered the most appropriate to implement the steps comprising the proposed learning thread.

MVC was first described in 1979 by Trygve Reenskaug [3] , then working on Smalltalk at Xerox PARC. The original implementation is described in depth in the influential paper "Applications Programming in Smalltalk-80: How to use Model–View–Controller" [5]. There have been several derivatives of MVC. For example, Model View Presenter is used with the .NET Framework [7], and the XForms standard uses a "model-view-controller-connector architecture" [1].

At the heart of MVC, the idea that was the most influential to later frameworks, is what it is called Separated Presentation. The idea behind Separated Presentation is to make a clear division between domain objects that model our perception of the real world, and presentation objects that are the elements we see on the screen. Domain objects should be completely self contained and work without reference to the presentation; they should also be able to support multiple presentations, possibly simultaneously. In MVC, the domain element is referred to as the model. Model objects are completely ignorant of the user interface. The presentation part of MVC is made of the two remaining elements: view and controller. The controller's job is to take the user's input and figure out what to do with it. It is important to highlight that there is not just one view and controller. There is a view-controller pair for each element of the screen, each of the controls and the screen as a whole.

A very common issue for the programmers is code reusing and refactoring. Many times we have to solve a problem similar to another one previously resolved, to improve the interface of a program, to optimize the underlying algorithm, etc. These tasks can be simplified separating the code into parts which can be reused without any modifications. Based on these premises, in order to reuse a code for future software development, the model must be independent. Model classes (or functions and structures) should not access any class from another group. This way we can compile the model in an independent library reusable for similar developments. Furthermore, if the software is developed using C programming language and we change the platform used (from example, from Linux to Windows), it should be possible to recompile the C code with minor modifications. So the model should not include any concrete graphics, sockets or other library calls which are usually very different, even within the same platform, if we use a different environment (think, for example, in graphics with Visual C++ and with Borland C++, both under PC/Windows).

Besides, the controller can (and usually does) access model and view classes by reusing code. View is the more changing element, so we link it to the controller classes. If we change the controller, at least we will have to recompile the graphics interface.

To get all this operating, the main function has to instantiate the model, controller, and view classes. In case a new graphics interface has to be developed, we will work without modifying the previously developed model and controller. It will be enough modifying the creation/instantiation of the graphic interface to include the new developed version. Everything should work without even recompiling the model and the controller.

IV. ITINERARY OF THE LEARNING PROCESS IN THE ALREADY ESTABLISHED THREAD

In order to perform a reusable project, a learning thread shared between several subjects was proposed. In a first implementation, it included two subjects: Computer Programming and Programming of Multimedia Applications. Performing the project should mean meeting both the final goal and the associated objectives. Each subject has its own partial goal (subprojects SP1 and SP2). In Figure 2 it can be observed the final architecture of the proposed project integrating the contribution of both subjects.

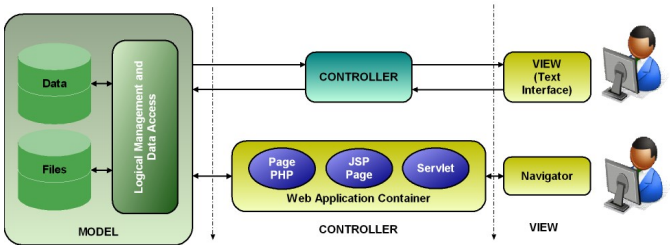

Figure 2. Architecture of the multidisplinary project to be developed in two subjects.

TABLE I. BREAKDOWN IN STAGES OF THE ESTABLISHED ITINERARY

Subject	Contribution to the thread	Result
Computer programming • 1st grade • 1st semester	Solving a complete problem using a high level programming language with a modular structure. Knowing how to specify and design modules and databases. Assembling a simple set of codes in a cooperative way using separate compilation units. Building a collection of function libraries.	SP1: A complete single-user application accessing to the application data following a model-view-controler (MVC) pattern. Text User Interface (TUI).
Programming of Multimedia Applications • 1st grade • 2nd semester	Solving same problem as previously presented with Web technologies. Developing a new graphic interface (view and controller) using HTML, CSS, JavaScript, and PHP programming language. Developing a new model if appropriate, or otherwise accessing to the previously developed model. Building new classes/modules (PHP) for accessing to the database (MySQL) into the model.	SP2: Adding to the application a new way of accessing, through Internet. User interface is a web site with dynamic pages. Controller are codified using a server language (PHP) and accessing to a database (MySQL).

MULTIDISCIPLINARY PROJECT: A TRAIN RIDE MANAGEMENT SYSTEM

As a specific example of a reusable project within the shared learning thread, we proposed last school year in the subjects Computer Programming and Programming of Multimedia Applications, a train ride management system. The project is divided into two different applications sharing same data. The first one, called *Management*, will be used by the travel agents to handle the information about train stations and rides. The second application, *TravelPlanning*, will be used by travel agents to provide the better travel plan to the clients according to their needs with the possibility to choose among the shortest, the fastest, and the cheapest travel. To compute the shortest path, the Dijkstra algorithm will be used.

A. Computer Programming

Students gain the abilities needed to solve a complete problem using a high level programming language with a modular structure, following the model-view-controller paradigm. They build a set of libraries in a cooperative way to perform mixed functions.

As a result, we obtain a complete single-user application (one process only) accessing to the application data by a hierarchical menu-based information retrieval system developed in a text user interface. Application execution is sequential and ruled by the main program controlling the evolution of the system states, particularly, the "dialogues" (input/output operations) which are always synchronous.

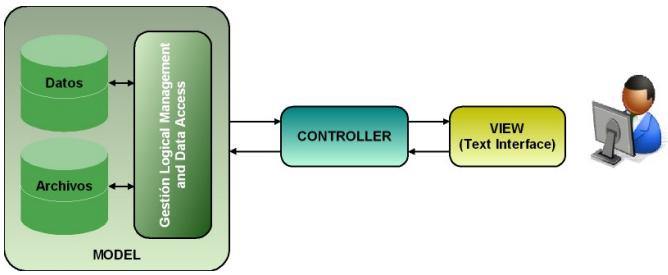

Figure 3. Architecture of the project to be developed in the Computer Programming subject.

MULTIDISCIPLINARY PROJECT RESULTS

In this first subject, the train routes planner problem is analyzed, designed, and implemented in two versions, using both C and C++ programming languages. In terms of the MVC paradigm, both applications use the same model and its view is a console application. In both versions, viewable result is exactly the same (see Figures 4 and 5).

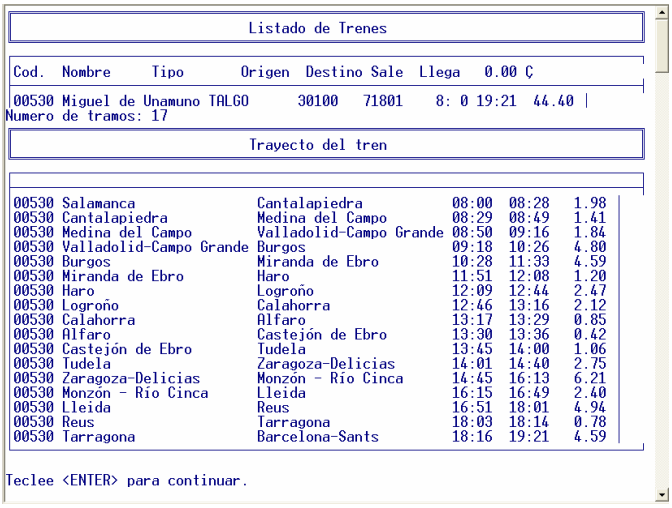

Figure 4. Screen shot of the *Management* application performed in the Computer Programming subject.

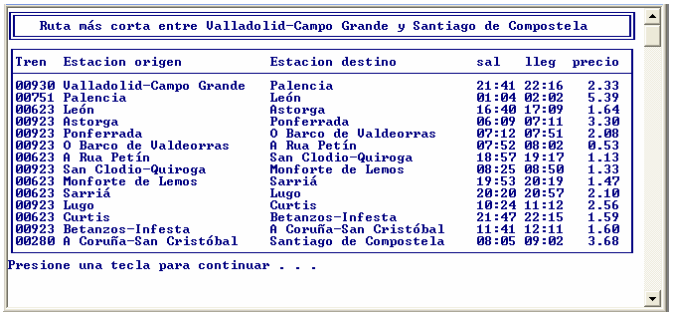

Figure 5. Screen shot of the *TravelPlanning* application performed in the Computer Programming subject.

B. *Programming of Multimedia Applications*

In this stage, the web technologies (HTML, CSS, JavaScript, PHP and SQL using MySQL DBMS) join to the learning thread. In this case, view will be a web page in a browser, which provides access from different platforms. Controller will be recoded using PHP and model (business logic) will be also recoded if necessary.

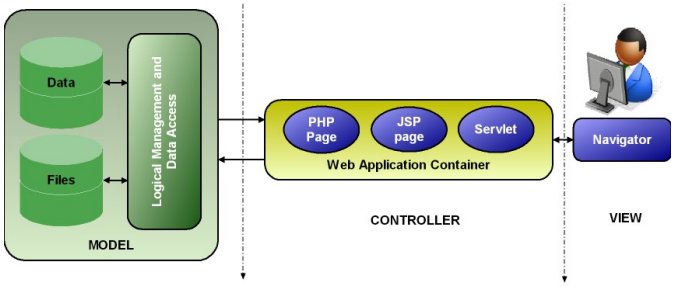

Figure 6. Architecture of the project to be developed in the Programming of Multimedia Applications subject.

As a result, we obtain a multi-user web application providing a user-friendly access from different platforms. This subject will disappear with the new grades, in its stead it will be lectured the subject Technologies of Web Applications ("Tecnologías de Aplicaciones Web"), where the web technologies can be widen.

MULTIDISCIPLINARY PROJECT RESULTS

The students start this subject using the first phase of the developed multidisciplinary project. The analysis and design steps have been previously done, so it is not necessary to prove abilities already evaluated in the Computer Programming subject. Within the MVC paradigm, the tasks to be performed to acquire the abilities covered by this subject are:

- Model: in the first version, data were stored in files. Now, a database will be added using MySQL.
- View: in the previous version, the user interacts with the system through a MS-DOS console. Now, a web site will be design to insert data and view the results obtained using the above-mentioned web technologies.
- Controller: the business logic was already designed and this design is language-independent. Now, the controller will be coded using the PHP language.

Therefore, the local application developed in the Computer Programming subject evolves into a network application with a graphic interface, more user-friendly than the console.

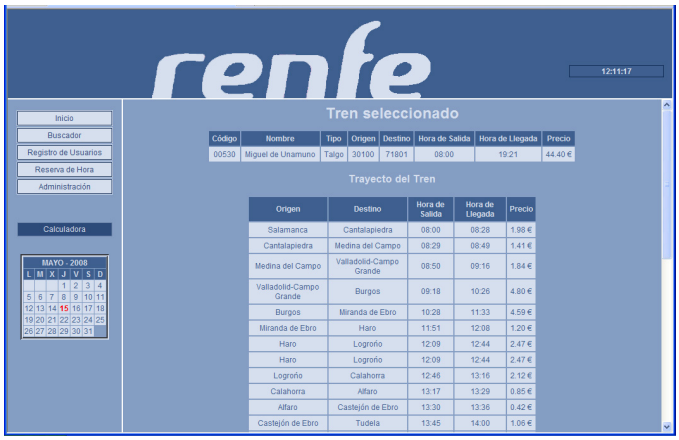

Figure 7. Screen shot of the *Management* application performed in the Programming of Multimedia Applications subject.

Figure 8. Screen shot of the *TravelPlanning* application performed in the Programming of Multimedia Applications subject.

V. ITINERARY OF THE LEARNING PROCESS IN THE THREAD PROPOSED

With the new grades starting, the Telecommunication Engineering degree will disappear while Telematic Grade will be established. Based on the gained experience and on the results obtained using the learning thread previously explained, a new learning thread is proposed with a multidisciplinary project along the four years of the Telematic degree. Hence, to better accomplish the final goal and the associated objectives, a five-step itinerary is proposed. Each step is tied to a subject, which has its own partial goal (subprojects SP1 to SP5). In the Figure 9 it can be observed the final architecture of the proposed project integrating the contribution of all the subjects.

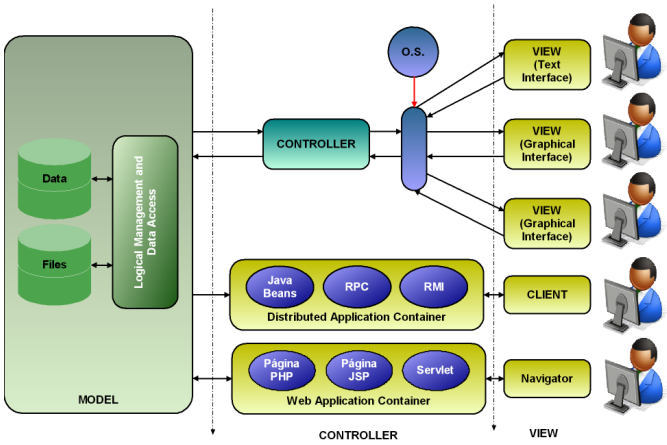

Figure 9. Architecture of the multidisplinary project to be developed along the degree.

TABLE II. BREAKDOWN IN STAGES OF THE ITINERARY PROPOSED

Subject	Contribution to the thread	Result
Computer programming • 1st grade • 1st semester	Solving a complete problem using a high level programming language with a modular structure. Knowing how to specify and design modules and databases. Assembling a simple set of codes in a cooperative way using separate compilation units. Building a collection of function libraries.	SP1: A single user complete application accessing to the application data following a model-view-controler (MVC) pattern. Text User Interface (TUI).
Computer Fundamentals and Operating Systems • 1st grade • 2nd semester	Incorporating concepts concerning to the operating systems to the project like the process planning, the process concurrency, and the communication between processes.	SP2: A complete multiuser application controlled by a dedicated operating system. The O.S. creates the semaphores and launches and controls the process controller and several view processes.
Software Systems Engineering • 2st grade • 2nd semester	Object-oriented programming. Software engineering. Event-oriented programming. Graphic interface.	SP3: Application is recoded using engineering software techniques. Controller is re-design using classes. A graphic interface is designed. *Database* access and management class library.
Architecture for Distributed Applications • 3st grade • 2nd semester	Client-server architectures. Sockets, middleware (RPC, RMI, EJB,…)	SP4: Adding to the application a new way of accessing: the remote access from other computer.
Technologies for Web Applications • 4th grade • 1st semester	Solving same problem as previously presented with Web technologies. Developing a new graphic interface (view and controller) using HTML, CSS, JavaSCript, and PHP programming language. Developing a new model if appropriate, otherwise accessing to the previously developed model. Building new classes/modules (PHP) for accessing to the database (MySQL) into the model.	SP5: Adding to the application a new way of accessing, through Internet. Use interface is a web site with dynamic pages. Controller are codified using a server language (PHP) and access to a database (MySQL).

Next, each subject contribution related to the learning thread will be explained. The contribution of the first and fifth subjects, already involved in the previous learning thread (Computer Programming and Technologies for Web Applications, earlier Programming of Multimedia Applications), are omitted as they were already explained.

A. Computer Fundamentals and Operating Systems

In this stage, new concepts over operating systems (O.S.), process planning, and communication among processes are built into the project.

The O.S. process of the application establishes the semaphores needed to control the concurrently access to the model of several clients and manages the communication between the controller and the view processes, applying different planning methods.

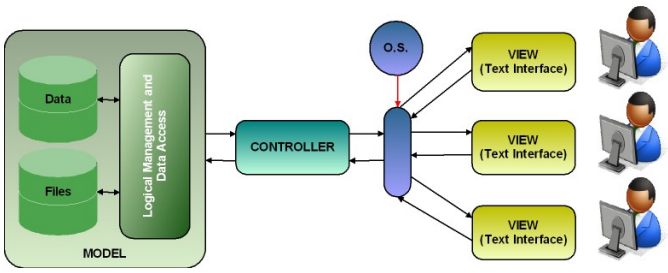

Figure 10. Architecture of the project to be developed in the Computer Fundamentals and Operating Systems subject.

As a result of the contribution of this stage, we obtain a complete multi-user and multi-process application managed by techniques used in real operating systems. In this stage communications among processes of the same computer are manage, which provides a background to the client-server architecture and the communications among process of different computers that will be study in next stages of the thread.

B. *Software Systems Engineering*

In this stage, methods from software engineering and application implementation are incorporated to the thread. The starting point is the analysis performed in the Computer Programming subject, then, the design of the required classes is performed. The model is encapsulated in a class library. The visual interface is designed following the event-oriented programming paradigm. Controller is also developed with classes.

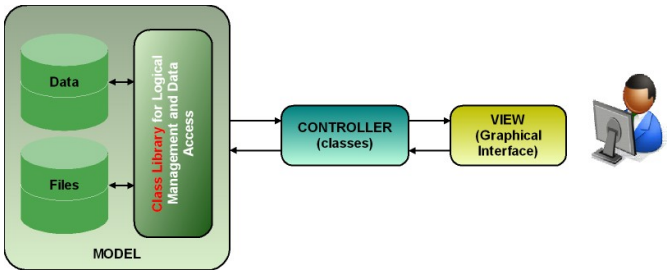

Figure 11. Architecture of the project to be developed in the Software Systems Engineering subject.

As a result of the contribution of this stage, we obtain a system completely implemented using classes, maintaining the model-view-controller scheme.

Including the result obtained in the Computer Fundamentals and Operating Systems subject, a system accessed from textual or graphic interfaces, or with the controller developed using structural or object-based programming language can be obtained. Furthermore, the function or the class library can be used, depending on the preferences or on the application to be developed.

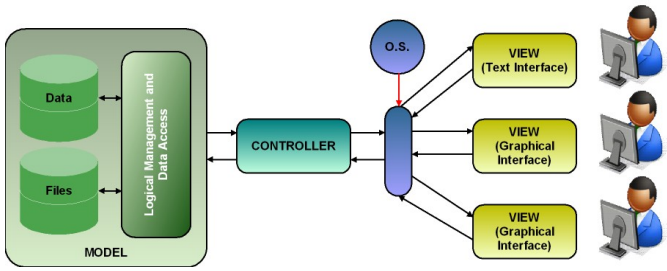

Figure 12. Architecture of the project to be developed in the Software Systems Engineering subject including the results obtained in the Computer Fundamentals and Operating Systems subject.

C. *Architecture for Distributed Applications*

In this stage, concepts related to remote communication among processes in distributed applications –sockets, middleware (PRC, RMI, EJB,…) are added, thus expanding the knowledge about communication among processes within the same computer provided in the Computer Fundamentals and Operating Systems subject.

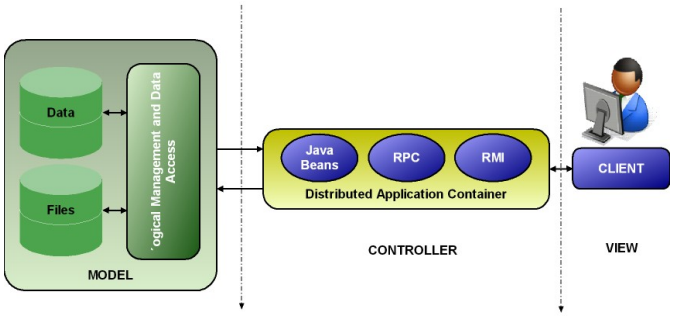

Figure 13. Architecture of the project to be developed in the Architecture for Distributed Applications subject.

As a result, an application is obtained keeping the model while controller is recoded, using specific technologies for distributed processing in different machines. View can be found in the same computer or in other.

VI. MOODLE: TOOL FOR LEARNING SUPPORTING

Some tools implemented as modules for the Moodle platform have been specifically designed to exploit the full potential of the PBL approach. A module called e-Liza is used for both a self-evaluation and a competitive evaluation of the gained knowledge. In e-Liza the questions are not only proposed by the teacher but also by the students. Of course, the last ones are supervised by the teacher accepting them or not, and assigning the difficulty level. Statistics of the individual or grouping play are shown to the students in order to keep them informed of their progress.

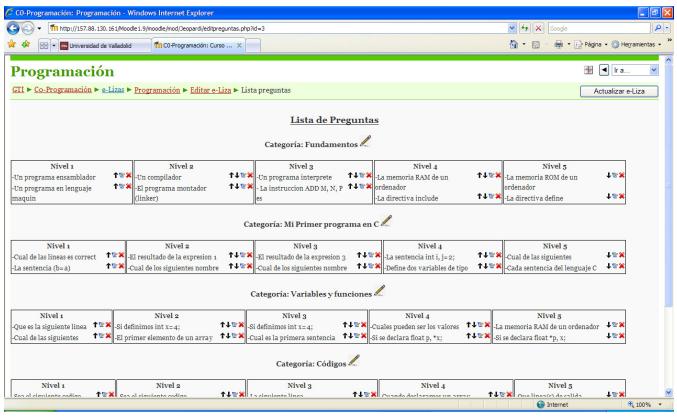

Figure 14. Screen shot of the module e-Liza, the section for entering the questions.

Another tool to perform questionnaires is used both for creating evaluation templates used in the peer-reviewing and for conducting surveys of the curriculum development. Students can also use other modules specially designed for each of the subjects, for example programming validators, also integrated in the Moodle platform. All these tools are very useful not only for the students making the learning process easier, but also for the teachers specially helping them in the classroom management, because this new approach adds more workload than the classical one, at least in their beginnings.

VII. CONCLUSIONS

In this paper a learning thread shared by several subjects of the Telecommunication Engineering degree related to application programming has been proposed. The goal is to place the student in a realistic scenario, facing real-world issues and following a pathway leading to the development of a global project throughout several years. Results obtained in the trials previously conducted are encouraging. Students get higher involved with the subjects and especially faster than without using this scheme. These facts provoke the students to get better marks, also decreasing the number of dropped out students.

As it has been explained, future work will consist of applying this methodology to a total of five subjects placed in consecutive semesters, coinciding with the new Telecommunication Engineering degrees beginning in the University of Valladolid.

REFERENCES

[1] Beltran Llera, J. Procesos, Estrategias y Técnicas de Aprendizaje. Editorial Síntesis, S.A. Madrid 1993.

[2] Boodhoo, J. P. Design Patterns: Model View Presenter. 2006.
http://msdn.microsoft.com/en-us/magazine/cc188690.aspx. Last visit: octobre, 2009.

[3] Burbeck, S. Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC).
http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html. Last visit: octobre, 2009.

[4] Clement, J. Model based learning as a key research area for science education. International Journal of Science Education, 22(9), pp. 1041-1053, 2000.

[5] Hadim, H. A. and S. K. Esche, Enhancing the Engineering Curriculum Through Project-Based Learning, 32nd ASEE/IEEE Frontiers in Education Conference, Boston, MA, 2002.

[6] Kolmos, A. Reflections on project work and problem-based learning, European Journal on Engineering Education, vol. 21, pp. 141-148, 1996.

[7] Trygve M. H. Reenskaug/MVC—XEROX PARC 1978-79.
http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html. Last visit: octobre, 2009.

[8] World Wide Web Consortium (December 9, 2008). The Forms Working Group. 2008.
http://www.w3.org/MarkUp/Forms/. Last visit: september, 2008.