BASICS OF WEB APPLICATION DESIGN
An Example-based Learning Approach

M. Gestal, D. Rivero, J. R. Rabuñal, J. Dorado and A. Pazos
Information and Communications Technologies Department, University of A Coruña, Spain

Keywords: Design patterns, Modeling, Web Applications, Case-Based Reasoning

Abstract: Traditionally, a great number of business applications were developed using the J2EE framework. This fact has led to the emergence of a multitude of tools, manuals, etc. that explain the different alternatives or features needed to implement them. The development of .NET Framework has resulted in a strong demand for the implementation of applications using this kind of architecture. However, the quantity or quality of the documentation available is far from the one available for J2EE. This detail is especially worrying when the main objective set is to reveal the concepts of framework from a teaching point of view. This paper describes the teaching approach followed to achieve the abovementioned aim, based mainly on a set of simple tutorials that show the technology basics and two complete applications (miniportal and minibank) in which the application of design patterns when dealing with a business application is presented.

1 INTRODUCTION

The subject of "Systems Integration" is a core subject in the 5th year of Computer Engineering. It lasts one year, comprising 15 credits.

Out of these, half of them correspond to teaching J2EE concepts and the other half to teaching .NET concepts (Cooper, 2002) (Gamma et al, 1995).

The subject examines both the technology and the most notable design techniques. During its development, a series of tutorials on the concepts necessary to put it into practice is taught: software model, patterns, technology, development environment, and so on. Moreover, a series of web applications are developed and made available to the students, as tutorials, in which the implementation of different concepts previously shown are detailed: user authentication, data validation, transaction management, etc.

This subject is essentially practical. Its assessment is determined by carrying out two web applications (one using J2EE technology and the other .NET), although it is also required to obtain a minimum grade in a multiple-choice test which verifies the proper assimilation of the concepts needed to carry out the practice tasks.

2 GLOBAL DESCRIPTION AND OBJECTIVES OF THE SUBJECT

The subject of "System Integration" focuses on the design and implementation of web applications with Java POJO and .NET technologies, with particular emphasis on the development of Web applications (Jacobson et al, 1999).

During the teaching period, practice tasks are developed, consisting of two Web applications, one implemented using J2EE and another using .NET. Both applications interact (through web services that return information in XML format) Thus, the .NET application can communicate and interact with the functionality exposed by the Java EE application. This work is the application in a real application (although clearly limited in terms of functionality due to time constraints) to the concepts shown during the teaching period.

The successful carrying out of the practice tasks allows the student to achieve targets set for the subject:

- Knowing the basics of programming using J2EE and .NET technology.
- Knowing the fundamental architectural principles of business applications.

Gestal M., Rivero D., R. Rabuñal J., Dorado J. and Pazos A. (2010).
BASICS OF WEB APPLICATION DESIGN - An Example-based Learning Approach.
In *Proceedings of the 2nd International Conference on Computer Supported Education*, pages 481-484
DOI: 10.5220/0002858504810484

- Knowing the basics of the interoperability of applications.
- Knowing design techniques in order to develop business applications (especially concerning Web applications) through a layered architecture.

3 TEACHING METHODOLOGY

In spite of having only practical credits, during the development of the subject, we also make use of lecture sessions. They consist of theoretical basic concepts, necessary to carry out the practice tasks. Thus, special emphasis is placed on database access technologies or software patterns used for the development of the model layers, view and controller of business applications.

Each part of the subject involves carrying out a practice task, to be conducted in groups. Each task tries to mimic the generic performance of known Web applications (Betandwin, Amazon, etc.), although obviously with fewer features due to the limited time available for its completion.

For each application two iterations or deadlines are defined. In the first iteration, which does not entail a grade, the initial part is implemented. The objective of this first iteration is to try to guarantee that the students focus on its development. To this end, the teacher tries to detect significant errors, and if there is any, the students are guided towards their solution. In the second iteration, the students correct the errors identified in the first one and add the other features.

As discussed above, the second application uses the functionality provided by the first one. In this way, the students will be helped to go in depth into the aspects relating to the interoperability of applications, one of the objectives of the subject.

Due to the extent of the practice task, the division of iterations to be handed in makes easier that a larger number of students reach the agreed deadlines and giving up occurs less often.

4 PROPORTIONAL IMPLEMENTATIONS

A very important part in the progress of the subject consists of the development and giving of a series of examples to students. These examples allow to observe the implementation of the theoretically explained concepts. This makes it possible to reduce the student's learning curve, as they can check the concepts required subsequently to perform practice tasks using functional examples.

The development of examples also facilitates the practical implementation since the example code is taken as a starting point for their progress. Thus, the students finally develop a complex (and perfectly usable) web application, without having to encode it entirely.

Two complete web applications are also provided (MiniPortal and MiniBank), in which the emphasis is placed on the fundamentals of business web application development. These applications involve different aspects (layering, authentication, transaction management, etc.) that subsequently, the students should apply for the completion of the practice tasks. Both applications are completed using .NET and Java.

As follows, a brief description of each of these materials is performed.

4.1 Tutorials

First, a series of short tutorials programs are developed and made available to students. These tutorials are intended to be seen as a set of examples, each of which shows the operation or performance of a particular aspect.

The tutorials are implemented considering they would be used as a technology learning guide, leaving aside issues such as efficiency or correctness of design. This approach comes from the fact that most of the technologies are new to the student, so we try to provide them in a way to be understood as far as possible. The tutorial sessions are organized as a set of examples, each focusing on the use of a particular aspect. All examples are thoroughly documented in order to facilitate their assimilation.

Regarding Java technology, students are given various examples of access to databases using JDBC, ORM Hibernate, as well as Tapestry framework.

Concerning the .NET technology, a similar approach is followed. Thus, a number of examples are given, which initially show basics of C# programming language (structures, exception handling, etc.). The tutorial session also includes examples of connection to databases through ADO.NET (on line and off line environment) and ORM Entity Framework.

4.2 MiniPortal

As discussed above, the MiniPortal application is a complete web application. Specifically, it is a portal

that allows registration and authentication of users, (see in Figure 1, the main business objects). This will be one of the requirements that the practice tasks of students must enclose, so that the example provided can be reused, with minimal changes.

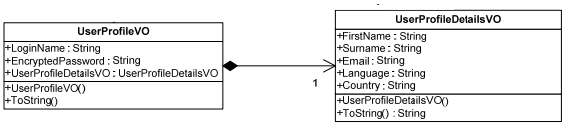

Figure 1: Domain objects in MiniPortal Example.

In the MiniPortal the students are shown in a practical way the division of a layered application (model-view-controller), and the use of certain design patterns (Session Facade, Business Delegate, Factory, etc.). The use cases (operations supported by the application) are shown in Figure 2.

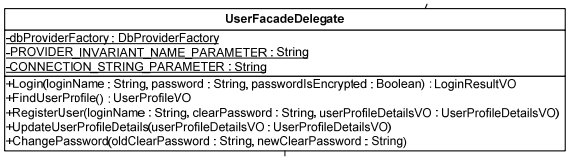

Figure 2: Facade with MiniPortal use-case's.

When implementing the model part, a special emphasis is placed on the management of data stored during the session (data to be accessed on different pages). Moreover, it is shown how to perform the test development (to this end, JUnit is used in the example developed in Java and a TestProject is used in the example developed in .NET), employment of externally adjustable parameters, etc.

On the website, several aspects are displayed: navigability between pages (see Figure 3), profile management, internationalization, passing parameters between pages, the model method calls, etc.

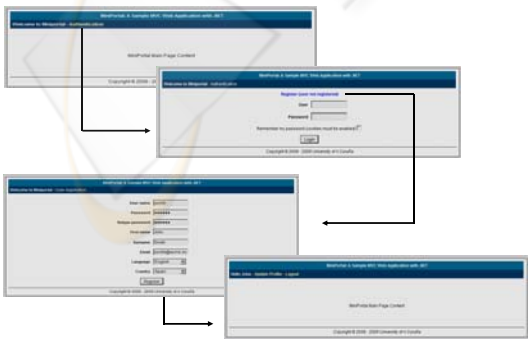

Figure 3: MiniPortal web interface: example of interaction with user registration.

4.3 MiniBank

The other fully developed example is aimed at showing the students how to simplify the bank account management (see in Figure 4 the main business objects).

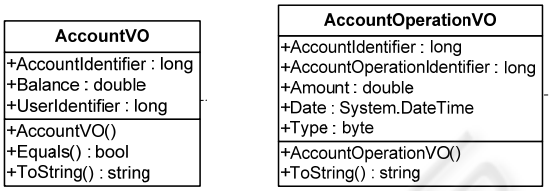

Figure 4: Domain objects in MiniBank Example.

MiniBank provides simpler (and more common) operations that can be performed on a bank account: creation, search, cash deposits and withdrawals or transfers between accounts (see Figure 4).

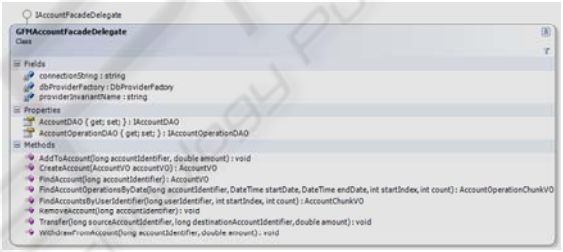

Figure 5: Facade with MiniBank use-case's.

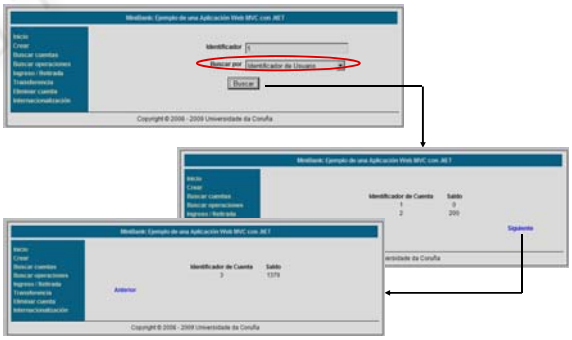

Figure 6: MiniBank web interface: example of interaction with account search operation and pagination results.

This example is intended to involve aspects not contemplated in the MiniPortal and that the students have to use when carrying out the practice task. In this way, aspects relating to the automatic generation of identifiers for storage in databases, transaction management or Page-by-Page iterator pattern (allowing the pagination of results when these are too high to be displayed at once) are shown.

In this example, the web interface is a bit simpler in the sense that the concepts already shown on the MiniPortal are highlighted. However, everything regarding the pagination of results is new (see fig. 6), as discussed above.

5 EVALUATION

As mentioned earlier, the subject (in terms of allocation of credits) is fully practical. therefore, its evaluation has to be obviously based on its practical aspects. To this end, after the delivery of each of the two practice tasks, their correction is proceeded to. this correction is carried out by the group through the defense of the presented practice task. this defense includes verifying the correct implementation of the practice task (therefore, a series of critical points are checked) as well as the group's proper understanding of the concepts used for the development of the practice task. according to the gravity of the errors detected, if there is any, each web application is assessed by a grade ranging from 0 to 5.

Additionally, two multiple choice examinations will be performed, one for Java POJO and another for .NET. The objective of multiple choice examinations is to verify whether the student has assimilated the concepts correctly. Each multiple-choice test consists of a set of questions with several possible answers, out of which only one is correct. No grade is awarded for unanswered questions, and one grade is deduced for each question answered incorrectly.

Requirements for passing the course: (1) having passed each of the two Web applications and (2) get at least 4.5 points (out of 10) in each of the two multiple-choice test. In principle, the final grade of a student who meets these two conditions is the one obtained as a result of the practice tasks (which is obtained as the sum of the grades of the two Web applications performed). However, the test grades may vary up or down from the grades to Web applications (although never to fail).

6 EXPERIENCE

According to surveys on students, the subject is among the most valued of the degree. And this occurs despite (or precisely due to it) of being one of the subjects in which the workload is higher. From our viewpoint, this is due to the fact that the students, since they carry out a real web application (although with reduced functionality) can see the fruit of their labor constantly, obtaining an application with which they are able to interact.

Concerning the examples carried out, we could say that their success is great because, when analyzing the logs located in the web servers, we can observe that the students' queries are only a small proportion of all the existing ones.

REFERENCES

Cooper, J.W. (2002). *C# Design Patterns: A Tutorial.* Boston; Addison-Wesley.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Toronto: Addisson-Wesley.

Jacobson, I., Booch, G., and Rumbaugh, J. (1999). *The Unified Software Development Process*, Boston: Addison-Wesley.

Web Resources Available at

Computer Science School. University of A Coruña. http://www.fic.udc.es

Subject Contents and Samples (.NET contents). http://sabia.tic.udc.es/docencia/is.

Subject Contents and Samples (Java contents). http://www.tic.udc.es/is-java/