From UML models to automatic generated tests for the dotLRN e-learning platform

Ana Cavalli [1]   Stéphane Maag [2]   Sofia Papagiannaki [3]
Georgios Verigakis [4]

GET / Institut National des Télécommunications
9 rue Charles Fourier
F-91011 Evry Cedex, France

Abstract

This paper presents a method for testing an e-learning, web based system. System specifications are provided using the UML modelling language and specifically the Sequence, Activity and Class diagrams of UML. These specifications are exported in XMI format which is parsed in order to produce the test cases. The system under consideration in this paper is dotLRN, an open source enterprise-class suite of web applications and portal framework for supporting course management, online communities and collaboration.

Keywords: UML, dotLRN, XMI, Component Testing, Validation

1 Introduction

During the last few years more and more organizations and companies exploit the services they provide by making them available through their web sites. In this context, educational organizations can adopt an e-learning application to manage and support remote distance courses. Among the advantages of such a course over a conventional one is its capability to overcome the geographical barriers and, as a consequence, to address to a larger audience. Furthermore,

---

[1] Email: Ana.Cavalli@int-evry.fr
[2] Email: Stephane.Maag@int-evry.fr
[3] Email: Sofia.Papagiannaki@int-evry.fr
[4] Email: Georgios.Verigakis@int-evry.fr

it is very important to make the cost of such a system is minimal. The organization only needs to deploy a powerful web server with the goal that the users may access these services using a minimum computing equipment connected to the internet.

Since software systems get more and more sophisticated and complicated, software testing is very important not only to uncover bugs but also to ensure that the system conforms with the specifications and really does what is supposed to do. The fact that big systems are decomposed in smaller sub-systems developed by different teams brings new challenges to testing, since it has to verify that these sub-systems are integrated together smoothly.

Furthermore, testing an e-learning system is essential because by its nature, such an application is exposed in many threats: a web server has to respond to request from each node in the network. Therefore, testing should guarantee that a user cannot gain permissions over the system he is not supposed to have. For instance it would be undesirable if a student got the permission to change his grades. In addition to this, modern web applications are becoming increasingly complex and mission critical. This situation becomes even worse because of the lack of standardization in the web clients (browsers). Testing has to assert the system usability. For instance a user is unable to complete a process because the content of the web page does not appear correctly or due to a slow network connection he may be unable to login because of not realistic timeout value. Our approach takes into account how these functional aspects have been implemented, checking that they conform the specification.

This paper is part of the work being developed in the framework of the E-LANE project, an European and Latin American collaboration for the creation of an advanced integrated system for e-learning in which GET/INT is a partner[5]. We propose a testing method for dotLRN, that is composed by a suite of web applications and a portal framework of an e-learning system, and is used as the base platform for the development of E-LANE.

Our methodology is mainly influenced by [1,2], which describes how UML diagrams developed in the Analysis phase are analyzed in an automated way in order to produce test cases. Indeed, while [2] focuses on integrating testing conducted in an incremental way, [1] describes how to use class invariants and a detailed formal description in UML diagrams. Both of them are used to derive test requirements and the test suites. In this paper, the purpose and the challenge is that instead of an object-oriented software we have to use UML to model and then to validate a web application tool. The methodology presented in these two previous papers are related to big and complex real-world systems and are not applicable to web-based applications. Indeed, here

---

[5] http://www.e-lane.org