MELIS: an Eclipse Based Environment for the Migration of Legacy Systems to the Web

Massimo Colosimo, Andrea De Lucia, Rita Francesc, Giuseppe Scanniello, and Genoveffa Tortora
Dipartimento di Matematica e Informatica – Università di Salerno
Via Ponte don Melillo – 84084 Fisciano (SA) - Italy
cillo_81@yahoo.it, {adelucia, francese, gscanniello, tortora}@unisa.it

Abstract

In this demonstration, we present MELIS (Migration Environment for Legacy Information Systems) an integrated environment for the migration of legacy systems to a multi-tier web-based architecture. This environment has been developed as an Eclipse plug-in to support the software engineer in the migration of the graphical user interface, and in the restructuring and wrapping of the original legacy code.

1. Introduction

Legacy Information Systems (LISs) are business critical and often operate 24 hours a day. They are written in some legacy language like COBOL, and their technical infrastructure is obsolete and does not support any kind of integration with web technologies. To remain competitive in the global market, organizations have to change their business providing services using web-enabled applications [1][2]. The LIS migration towards a new technological infrastructure represents a good opportunity [1] to preserve the past investments reducing risks and costs.

In this demonstration we present MELIS (Migration Environment for Legacy Information Systems) an integrated environment implemented as Eclipse plug-in for the migration of LISs to multi-tier web-based architectures [2]. This plug-in supports the software engineer in the migration of non decomposable multi-user systems written in ACUCOBOL-GT programming language, a COBOL dialect.

2. The Migration Process

To migrate the original LIS to the web we adopted the target architecture and the migration strategy proposed in [2]. The target architecture (see Figure 1) integrates the Wrapper to enable the communication between the Reengineered GUI and the application logic of the LIS, which has been properly restructured (Restructured LIS). The Wrapper starts the Restructured LIS and manages the synchronization of the executions of the LIS and the new web based user interface. To this end this component provides two different interfaces. The first interface is provided to the LIS which is turned into a batch program. The second interface is provided to the GUI Deliverer software component implementing the new web user interface to access the functionalities of the LIS. The GUI Deliverer invokes the Wrapper interface using either RMI or SOAP. The Wrapper is a generic component that can be used in the migration of any LISs.

![Figure 1. The target software architecture](image_url)
It is important to point out that the migration of the legacy user interface has to be performed in a semi-automatic way because of the presence of embedded control flow in the user interface description. Indeed, SCREEN SECTIONs contain BEFORE and AFTER code that cannot be migrated to the client side because they contain checks involving database accesses (READ/WRITE statements).

3. The Migration Environment

MELIS is an integrated environment [2], which has been designed and implemented to facilitate the comprehension and the migration of the LIS graphical user interface to the web respecting the target architecture and the migration strategy described above. The environment enables us to automatically produces an XML file describing the structure of the SCREEN SECTIONs of the COBOL program. The XML file includes the actions associated to a SCREEN SECTION, which has to be performed when the entry fields are filled in and the confirm button is pushed. Moreover, for each identified SCREEN SECTION the XML file contains details of the corresponding graphical objects and the associated controls. The XML file is used to hierarchically organize the code associated to the SCREEN SECTIONs. The first level of the hierarchy reports the names of the SCREEN SECTIONs, while their graphical objects and the corresponding BEFORE and AFTER clauses are shown at second and third level, respectively. These clauses are represented as leaf nodes of the hierarchy together with the names of the associated procedures. Generally, controls in terms of BEFORE and AFTER clauses can be migrated either on server or on the client side. We use the javascript language to migrate on the client side checks that do not need any interaction with the LIS, as for example the validation of the date format. MELIS also proposes a library containing a set of javascript functions that can be reused during the client-side control migration.

The source code of checks involving database accesses cannot be migrated to the client. The plug-in analyzes the legacy code and highlights it in yellow when the checks can be migrated on the client side, in red otherwise. During the Restructuring and Wrapping phase the statements DISPLAY and ACCEPT are automatically commented (through a “*”) and then replaced by calls to the Wrapper component. Transitions between SCREEN SECTIONs are preserved in the reengineered graphical user interfaces as well as the error handling. An ACUCOBOL-GT compiler has also been integrated within MELIS to re-compile the restructured ACUCOBOL-GT program.

During the GUI reengineering phase the XML file enhanced during the whole restructuring process is successively used to generate a JSP page (see Figure 2) for each SCREEN SECTION of a given ACUCOBOL-GT subsystem. To enable the communication between each JSP page and the restructured legacy code a servlet is also automatically generated. The generated pages are very similar to the original one, but they can be eventually modified or enhanced by using the authoring feature of our environment. For example, Figure 2 shows a JSP page generated by our environment and then enhanced to reproduce the layout of the original user interface. Moreover, we also adopt the Cascading Style Sheets (CSS) to obtain the more appropriate look and feel and to modify easily the reengineered user interface once the restructured LIS has been integrated and deployed. The web based application, i.e., the generated beans, servlets, and dynamic pages, has to be deployed on the Server node. To this aim, MELIS allows starting Tomcat inside Eclipse by using the Eclipse Web Tool Platform (WTP).

4. Conclusion

In this demo we mainly focus on the main features of the proposed Eclipse plug-in. Currently, we are using MELIS to support a migration project funded by MTSys S.r.l., an Italian small software enterprise.

References