An Eclipse-based Tool for Model Driven Web Applications with Personalization Support

Irene Garrigós[1]

Octavio Glorio[1]

Paul Hernández[1]

Alejandro Mate[1]

Abstract: Web engineering methods provide a systematic approach to develop complex and high quality Web applications. Many of these methods provide, in some way, personalization support. However, none of the existing Web methodologies provide a tool which allow the modeling, automatic generation and deployment of personalized Web applications in a complete way. In this line, they also do not exist tools which allow the designer modifying the personalization strategies at runtime, avoiding to regenerate Web applications from scratch. To tackle these lacks, in this paper we have presented a tool that has been developed to give support to the A-OOH (Adaptive Object Oriented Hypermedia) design method. The main advantages of this tool are the automatic generation and deployment of personalized Web applications with personalization support at runtime.

1   Introduction

The main objective of the personalized Web applications is to improve the user satisfaction by providing him with a customized view to satisfy his needs and goals when browsing the website. In this way, the content, navigation and presentation of the Web application can be adjusted to the user specific requirements and preferences, easily finding what he is looking for.

1 Lucentia Research Group. Department of Software and Computing Systems, University of Alicante, Spain
  {igarrigos,oglorio,phernandez,amate@dlsi.ua.es}

Web modeling methodologies provide a systematic approach for the development of Web applications, and many of them support in some way the modeling of the personalization. However, none of the existing Web methodologies provide a tool which allow the modeling, automatic generation and deployment of personalized Web applications in a complete way. In this time, there also not exist tools which allow the designer modifying the personalization strategies at runtime, avoiding to regenerate Web applications from scratch.

As an example, we can mention the tool associated with the Hera Web modeling approach (HPG) [1], in which visibility conditions are specified over the design modeling elements of Hera. This tool does not automatically install the generated Web application, and it also does not allow the modification of these visibility conditions at runtime, because they are embedded in the Web pages programming code. Another example is WebRatio [2] developed to give support to the WebML methodology, it only considers the specification of static personalization (i.e. defining different views at design time) with respect to the user characteristics or browsing device. Therefore, the generation of such applications is similar to the generation of a traditional Web application (without personalization support).

The tool that is presented in this paper has been developed to give support to the A-OOH (*Adaptive Object Oriented Hypermedia*) [3] design method. This approach is the extension of the OO-H [4] modeling method, with personalization support. The main objective of this tool is the automatic generation of personalized Web applications from conceptual models (defined with A-OOH) and its further deployment. Other of the main goals is the maintenance of the personalization at runtime.

The fundamentals of this tool were published in [5]. Nowadays, this tool has been improved adding graphical editors for the models, and allowing an automatic installation of the generated personalized Web application and the needed modules. Moreover, this tool includes the possibility of defining a requirements model for the Web application presented in [6], which final goal is the automatic generation of a first skeleton of the conceptual models that the designer will have to refine.

2   Model Driven Development of Personalized Web Applications

Typically, to define a Web application we have to define three main design models: a *domain model,* in which the structure of the domain data is defined, a *navigational model,* in which the structure and behavior of the navigation view over the domain data is defined, and finally a *presentation model,* in which the presentation of the application is defined. To model the personalization at design time, two additional models are needed: a *personalization model,* in which the personalization strategies are specified, and a *user model,* in which the structure of the needed information for personalization is specified.

In [6] it has been added to the set of models of A-OOH, a *requirement model* based in i* [7] for the specification of user requirements. This model can be defined in the presented tool. With the objective that the requirements are consistent with the design it is

intended to perform model transformations to generate tentative conceptual models of the Web application. We have defined a set of transformation rules for certain conceptual models, however this transformation module is not complete, therefore it is outside the tool.

The presented tool takes as *input* the set of A-OOH design models, in XMI [8] format and the generation engine obtains the following *output:*

- Web application: a set of Web pages in ASP.NET is generated

- Application database: the conceptual models are mapped to an objetct-oriented database.

- Modules to manage the personalization: specifically a Web engine to manage the events, a module to evaluate and execute the personalization rules modifying the conceptual models and generating the needed parts or page(s) and a module to manage the personalization at runtime.

2.1 Implementation

The proposed tool has been integrated in the ECLIPSE[2] platform as it is shown in Fig. 1. Eclipse is an open source Project conceived as a modular platform able to be extended by plugins in order to add features to the development environment. In such way, we have developed a set of plugins that allow us designing conceptual models and upon them obtaining a personalized Web application.

The personalization model is built up by a set of Event-Condition-Action rules. It is defined by using PRML (*Personalization Rules Modeling Language*) [9] language in an orthogonal way to the rest of the models. These rules are saved in a text file separated from the application logic in order to easily modify them.

In order to give support to modeling tasks, we have used the UML (*Unified Modeling Language*) profile extension mechanism and we have used them in different diagram editors. Those editors have been built using the Eclipse native framework for modeling (EMF) and the graphical extension (GMF). On the other hand, in order to give support for code generation, we have developed another Eclipse plugin. This plugin is mainly made by a Web service (*Java*) and a generation engine for C# code derivation.

Finally, in order to give support to the generated Web application installation, we have developed an installer that configures and uploads the application in the server. Once the application is operative, the personalization rules could be updated during runtime in another integrated editor.

In Fig. 1, it is shown a snapshot of the developed tool. It is possible to observe different design models inside their respective editor and the personalization rules editor. In the upper side it is the tool menu, where it is possible to control all the developing process (code generation, application installation and personalization rules updating).
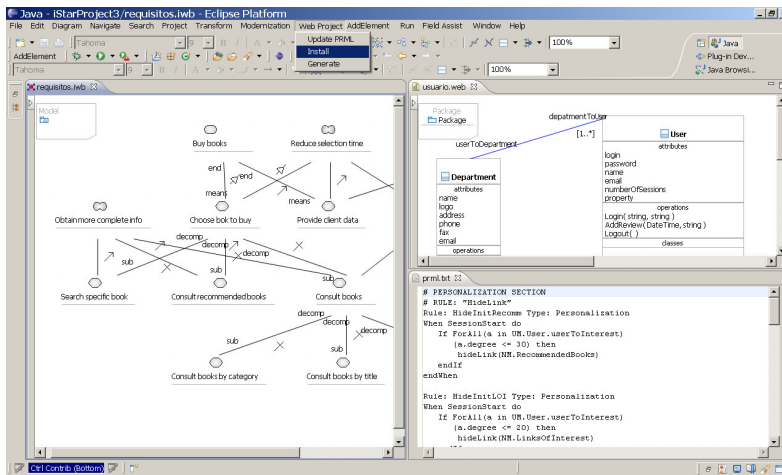
---

[2] URL: http://www.eclipse.org

Fig. 1. Developed tool snapshot

References

[1] Franciscar, F., Houben, G.J., Barna, P.: Hera presentation generator. In: WWW (Special interest tracks and posters). (2005) 952-953

[2] Acerbis, R., Bongio, A., Butti, S., Ceri, S., Ciapessoni, F., Conserva, C., Fraternali, P., Tofetti, G.: Webratio, an innovative technology for web application development. In: ICWE. (2004) 613-614

[3] Garrigós, I.: A-OOH: Extending Web Application Design with Dynamic Personalization. PhD thesis, University of Alicante, Spain (2008).

[4] Cachero, C., Gómez, J.: Advanced conceptual modeling of web applications: embedding operation interfaces in navigation design. In: JISBD. (2002) 235-248

[5] Garrigós, I., Cruz, C., Gómez, J.: A prototype tool for the automatic generation of adaptive websites. In: AEWSE. (2007)

[6] Garrigós, I., Mazón, J.N., Trujillo, J.: A requirement analysis approach for using i* in web engineering. In Gaedke, M., Grossniklaus, M., Diaz, O., eds.: ICWE. Volume 5648 of Lecture Notes in Computer Science., Springer (3009) 151-165

[7] Yu, E.: Modeling Strategic Relationships for Process Reenginering. PhD thesis, University of Toronto, Canada (1995)

[8] XML Metadata Interchange: http://www.omg.org/technology/documents/formal/xmi.htm

[9] Garrigós, I., Gómez, J.: Modeling user behaviour aware websites in PRML. In: WISM. (2006).