WebTE: MDA Transformation Engine for Web Applications

Santiago Meliá[1], Jaime Gómez[1], and José Luís Serrano[2]

[1] Universidad de Alicante, Spain
[2] Caja de Ahorros de Alicante, Spain
santi@dlsi.ua.es, jgomez@dlsi.ua.es, jlserrano@cam.es

Abstract. Transformations are of crucial importance for the success of Model-Driven Web Engineering (MDWE) approaches. Therefore, we need transformation engines to improve and obtain the results of the different approaches. However, very few model-driven Web approaches provide a transformation tool which would allow them to obtain an implementation from their models. In this paper, we present a tool called WebTE (WebSA Transformation Engine) which is able to introduce all the input artefacts of the WebSA approach and to establish a refining process based on model-to-model and model-to-text transformations which gives us the final implementation of a Web application.

1 Introduction

Model-Driven Engineering (MDE) is becoming a widely accepted approach for Web Engineering discipline. In fact, existing model-based Web engineering approaches currently provide excellent methodologies focused on the functional aspects. However, due to the novelty of this discipline, there are some aspects which have not been tackled yet: (1) the architectural aspects which would permit to obtain some quality attributes such as distributed computation, scalability, maintenance, connectivity with legacy systems, etc. (2) Traceability from the design models to the implementation. (3) The existence of too many notations to represent the same functional concepts in the different Web methods. (4) Currently, there is not any tool which allows us to define a complete model-driven approach based on transformations to obtain the final implementation.

To overcome these limitations, WebSA [7] defines a specific development process for Web Applications which regards software architecture artefacts as first-class citizens introducing automation mechanisms that accelerate this process. To do so, this approach defines a set of architectural models which complements the functional models of other Web methodologies providing a complete specification of the application. From these models, this approach starts an automatic and traceable process through a set of model-to-model transformations which carries out the integration of the architectural and functional aspects in a design model. In the last step, the process defines a set of model-to-text transformations which converts the integrated design model into different platform implementations. In order to give support to this approach, we have implemented a Web Tool called WebTE [12] which is basically a

L. Baresi, P. Fraternali, and G.-J. Houben (Eds.): ICWE 2007, LNCS 4607, pp. 491 – 495, 2007.

Web application which, through Web pages, allows us to introduce WebSA's models and transformations into a transformation engine which will execute them and give us the result of the WebSA process.

Section 2 gives an overview of the WebSA development process. Section 3 presents the specification of the WebTE tool and its user interface. Finally, section 4 presents some future steps and the conclusions.

2 The WebSA Approach: An Overview

WebSA [7] is a proposal whose main objective is to cover all phases of Web application development focusing on software architecture. It contributes to fill the gap currently existing between traditional Web design models and the final implementation. In order to achieve this, it defines a set of architectural models to specify the architectural viewpoint which complements current Web engineering methodologies.

The WebSA development process is based on the MDA development process [4], which includes the same phases as those included in the traditional life cycle of an application (analysis, design and implementation). However, unlike in the traditional life cycle, in the MDA approach [10] the artefacts that result from each phase must be models, which represent the different abstraction levels in the system specification. In the analysis phase the Web application specification is vertically divided into two viewpoints. On the one side, the functional-perspective is given by the Web functional models provided by approaches such as WebML [2], OO-H [3] or UWE [6]. On the other side, the Subsystem Model (SM) and the Configuration Model (CM) define the software architecture of the Web Application. Defining the application architecture orthogonally to its functionality allows for its reuse in different Web applications. The models-to-model transformation which goes from analysis models to platform independent design model provides a set of artefacts in which the conceptual elements of the analysis phase are mapped to design elements. It integrates the information about functionality and architecture in a single Integration Model (IM). This transformation type will be called T1 in the rest of the article. The Integration Model is the basis on which several model-to-text transformations, one for each target platform, can be defined. The output of these mode-to-text transformations is the implementation of the Web application for a given platform. This transformation type will be named T2 in the rest of the article.

3 WebTE: WebSA Transformation Engine

To give a support to the WebSA approach, we have implemented a Web tool called WebTE (WebSA Transformation Engine) [12] which is developed using the J2EE platform. The WebTE tool is based on the standards provided by the OMG (UML, XMI, MOF and OCL) mainly for optimizing the implementation effort, through off-the-shelf components and facilitating the use of any type of UML tool that possesses the support for class diagrams.

The main characteristic of the WebTE Tool is that it is a Web application and can therefore be used in a remote way. This is due to the fact that this tool does not need to provide a modelled graphic interface, because models and transformations are specified from any UML tool and it is from them that their representation can be generated in XMI. Besides, the text transformations are sent using text files.

Fig. 1 describes a complete process execution of the WebTE application, representing the different artefacts and components, each of which performs a task within the WebSA process. We will start the description in a gradual form using numbers that indicate the order of the process from source models to implementation.

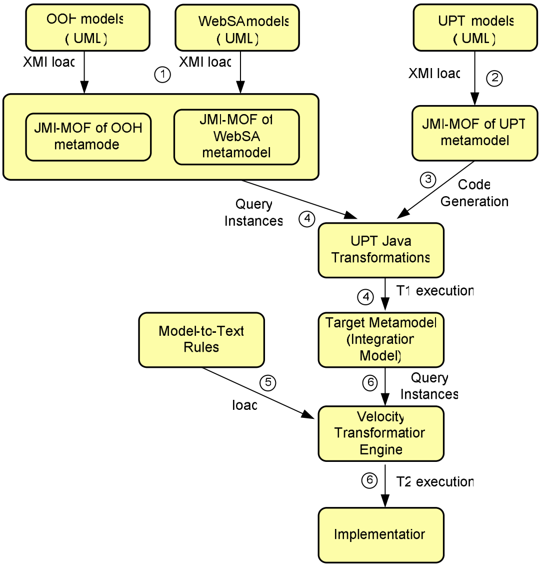
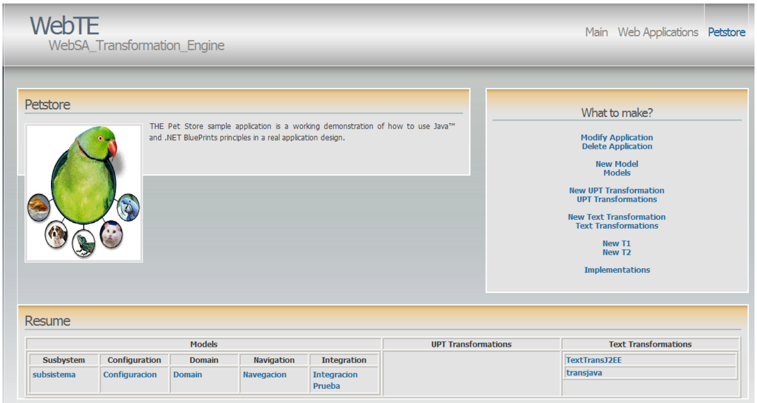


Fig. 1. The process of the WebTE Tool

It consists of the following steps:

1. The Web modeller expert defines the UML models of WebSA and the OO-H functional Web method using a UML tool, and then he/she generates the XMI representation which is loaded by the XMI parse of the JMI metamodels. The MOF metamodels of WebSA and the Web approaches are defined in JMI (Java Metamodel Interface) [4]. The use of JMI has two advantages: (1) it manages the instances of the MOF metamodel classes, that is, it makes it possible to manage models which are compliant with the metamodels, and (2) it provides the parses for reading and writing from XMI documents of UML that represent the models.
2. The Transformation expert establishes the UML transformations using the UPT language [8] in any UML tool capable of generate XMI documents of UML. The tool imports the XMI of the UPT transformations and from them creates the instances of the UPT metamodel.
3. It initiates a process of compilation in order to obtain the transformation in Java code and its subsequent compilation. This step is optional, because once the WebSA transformations have been generated, the rest of users can reuse them. However, in some cases, the user could be interested in introducing an extension of the WebSA transformations for a specific Web application.

4. The T1 transformation (models-to-model) is launched to convert the instances of JMI origin metamodels (WebSA and OOH) into the instances of the JMI target metamodel that correspond to the WebSA Integration Model.
5. The last phase of the WebSA process starts at this point with the T2 transformation which transforms the Integration Model into the platform implementation using model-to-text rules. To do this, the WebSA model-to-text transformation rules are written and loaded into the WebTE tool using text files. This step is also optional, because the WebTE tool has a complete set of predefined model-to-text rules for the most important platforms such as J2EE and .NET. However, a user could introduce new rules that are specific for his application.
6. Finally, the T2 transformation is executed and the instances of the WebSA metamodel are queried to generate the implementation of a specific platform (J2EE or .NET).



THE Pet Store sample application is a working demonstration of how to use Java™ and .NET BluePrints principles in a real application design.

Models					UPT Transformations	Text Transformations
Susbystem	Configuration	Domain	Navigation	Integration		TextTransJ2EE
subsistema	Configuracion	Domain	Navegacion	Integracion Prueba		transjava

Fig. 2. Main Page of WebTE for Petstore Web Application

Fig. 2 depicts a snap-shot of the WebTE tool. This Web page presents the main page of the well-known Web application Petstore [11] where we find an image of the application, a description and a resume of its artefacts (models, transformations and executions) that have been stored in the WebTE tool up to now.

Besides, on the right side, we can see a table which has a set of links with all the different actions that a user can perform using the WebTE Tool such as: *Modify Application* and *Delete Application* which allow us to manage a created application. *New Model and Models* permit to create and query the models, respectively. *New UPT Transformation and UPT Transformation* create and query the new model-to-model transformations specific for the Petstore application. *New Text Transformation and Text Transformation* create and query the new model-to-text transformations specific for the Petstore application. *New T1* starts the execution of the WebSA T1 transformation as does *New T2* for the WebSA T2 transformation. Finally, the *Implementation* link permits to query the set of Petstore implementations obtained up to now.

Up to now, WebTE only supports the Web functional models defined in the OO-H approach. However, WebTE could be easily extended to other functional approaches using the Netbeans MDR [9] framework which allows us to obtain automatically a JMI metamodel from a XMI-MOF metamodel defined in any UML tool.

4 Conclusions and Future Works

In this paper has been presented a Web tool called which WebTE is based on three main aspects: (1) support of the models and transformations of the WebSA approach. (2) Extensibility for adding and modifying models and transformations rules specific for a user or a system. (3) Interoperability between different tools in order to obtain different artefacts from them using XMI. WebTE offers a great tool compatibility because the models and transformations can be represented in any UML tool and can also be shared using XMI.

Today, this tool has been applied to an MDA proposal as WebSA and a Web Functional Method as OO-H but it could be applied to any approach which defines its metamodel in the standard MOF. In our future work, we will try to increase the number of possible users of the WebTE tool introducing other Web methodologies metamodels such as UWE [6], WebML [2] and W2000 [1]. It allows us to introduce their functional aspects into the WebSA development process.

References

1. Baresi, L., Garzotto, F., Paolini, P.: Extending UML for Modeling Web Applications. In: Proceedings of the 34th International Conference on System Sciences (2001)
2. Ceri, S., Fraternali, P., Matera, M.: Conceptual Modeling of Data-Intensive Web Applications. IEEE Internet Computing 6(4), 20–30 (2002)
3. Gómez, J., Cachero, C., Pastor, O.: Conceptual Modeling of Device-Independent Web Applications. IEEE Multimedia 8(2), 26–39 (2001)
4. Java Metadata Interface (JMI) (2006), http://java.sun.com/products/jmi/
5. Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture, Practice and Promise. Addison-Wesley, London, UK (2003)
6. Koch, N., Kraus, A.: The expressive Power of UML-based Web Engineering. In: 2nd IWWOST'02, CYTED. June 2002. pp. 105–119 (2002)
7. Meliá, S., Gomez, J.: The WebSA Approach: Applying Model Driven Engineering to Web Applications. Journal of Web Engineering, © 5(2), Rinton Press, 121–149 (2006)
8. Meliá, S., Gomez, J.: UPT. A Graphical Transformation Language based on a UML Profile. In: Proceedings of European Workshop on Milestones, Models and Mappings for Model-Driven Architecture (3M4MDA 2006). 2nd European Conference on Model Driven Architecture (EC-MDA 2006) (July 2006)
9. Metadata Repository Project HOME (MDR) architecture.html (2006), http://mdr.netbeans.org/
10. OMG. MDA Guide, OMG doc. ab/2003-05-01
11. TM J2EE Blueprint. Java Petstore 1.1.2 (November 2004) http://developer.java.sun.com/developer /releases/petstore/petstore1_1_2.html
12. WebTE., http://ebusiness.dlsi.ua.es:8080/WebSATool/