UWE4JSF: A Model-Driven Generation Approach for Web Applications

Christian Kroiss[1], Nora Koch[1,2], and Alexander Knapp[1]

[1] Ludwig-Maximilians-Universität München, Germany
[2] Cirquent GmbH, Germany
{kroiss,kochn,knapp}@pst.ifi.lmu.de

Abstract. Model-driven engineering is a promising approach, but there are still many hurdles to overcome. The tool UWE4JSF solves the hurdles for the model-driven development of web applications designed with UWE. It builds upon a set of models and domain specific annotations – in particular an abstract and a concrete presentation model. It is completely integrated in Eclipse, implemented as a set of plugins supporting model transformations and fully automatic code generation.

1 Introduction

The aim of model-driven development (MDD) is to raise the level of abstraction at which software is developed in order to save time and to reduce the amount of redundant programming work. MDD approaches are based on models that become first-class citizens in the development process, and on metamodels and model transformations requiring appropriate tool support. UWE4JSF [2] is such a CASE tool that was developed for the generation of web applications within the scope of the UML-based Web Engineering approach (UWE)[1].

UWE4JSF focuses on the automated generation of web applications, similarly to UWEATL [1] – first MDD approach for UWE – but differs in several conceptual and implementation aspects. In particular, (1) UWE4JSF is integrated in the Eclipse IDE using Eclipse-based transformation technologies. (2) It automatically generates web applications for the JSF[2] platform, a component-based technology which provides a flexible and powerful mechanism for the implementation of user interfaces (UI) of arbitrary complexity by means of component libraries. (3) UWE4JSF makes use of OGNL[3] that is an open-source expression language for Java. (4) The generation of the UI is based on a revisited version of the UWE presentation metamodel and an additional *concrete presentation model*.

To summarize UWE4JSF provides a human-readable, debuggable and high-performance approach that supports fully automated generation of web applications.

[1] UWE — http://www.pst.ifi.lmu.de/projekte/uwe
[2] Java Server Faces — http://java.sun.com/javaee/javaserverfaces/
[3] Object-Graph Navigation Language — http://www.opensymphony.com/ognl/

M. Gaedke, M. Grossniklaus, and O. Díaz (Eds.): ICWE 2009, LNCS 5648, pp. 493–496, 2009.

2 Extending UWE for Model Driven Development

UWE follows the principle of "separation of concerns" by modeling the content, the navigation structure, the business processes, and the presentation of a web application separately as shown in Fig. 1. UWE is mainly based on standards, like UML and MDA[4]. The models are built using the UWE profile, which is a UML extension defined using the extension mechanisms provided by UML. For example, classes with a stereotype «navigationClass» represent navigable nodes for information retrieval; associations stereotyped with «navigationLink» model direct links. For more details the reader is referred to [2].

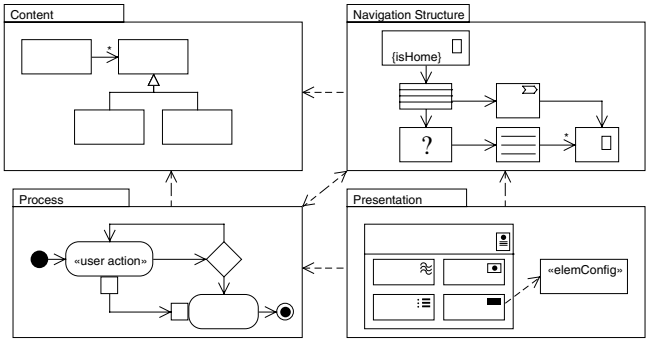

Fig. 1. UWE model types: overview

To use an UWE model for automatic transformation and code generation, it has to be augmented with explicit information that is not necessary if the model is only used for communication or documentation purposes. For example, the input data for navigation nodes has to be specified, together with the selection rules that are executed when links are followed. Information like this is specified by means of OGNL, combined with some UWE-specific functions. OGNL is also used in the activities of the process model to specify guard expressions and data handling actions. Unlike many other approaches, UWE also provides a UML extension for the explicit modeling of the user interface. First of all, the *(abstract) presentation model* is represented in UML using composite structure diagrams containing stereotyped classes and properties as representations for UI elements like text input fields or buttons. The resulting diagrams are very well suited to illustrate the basic layout and functional structure of the UI.

The elements of the presentation model must be mapped to UI components of the target platform. Many MDD approaches like WebML/Webratio[5] use template-based mechanisms for this purpose. However, in modern web applications, the selection of concrete UI components often strongly affects the usability, e.g. a date could be entered in a text field or with a dynamic calendar component.

[4] OMG — MDA Guide, http://www.omg.org/docs/omg/03-06-01.pdf
[5] http://www.webml.com

Therefore, in UWE this mapping is regarded as an important part of the application's design and recorded in a dedicated UML-based model, called the *concrete presentation model*. The basic idea is that platform specific UI components are modeled as stereotyped UML classes and corresponding instance specifications represent *concrete component configurations*. An example for a component that requires configuration like this might be a rich table whose column headers can be clicked to switch between different sorting criteria – the latter would be defined using attributes of the component configuration. It is also possible to build composite tree structures of component configurations, which might be used, for example, to add labels or headers to input or output elements. The mapping of abstract presentation model elements to element configurations can be established in two ways: (1) by associating mapping rules to meta-classes in *default element configurations* or (2) by linking individual elements of the abstract presentation model to element configurations with UML dependencies. In the sense of the MDA, these individual mappings could be seen as *markings* that guide the transformation process.

3 Tool Chain for Automatic Generation

The concepts described above were realized in the transformation and code generation tool UWE4JSF[6]. Its applicability has been demonstrated with several example applications, e.g. a simplified MP3 web store. UWE4JSF is implemented as a set of Eclipse plugins and supports automatic generation of JSF-based web applications from UWE models as well as model validation using constraints specified in the Object Constraint Language (OCL). UWE4JSF uses EMF[7] for the storage of (meta-)models and for the exchange with third party UML CASE tools, using a widely supported data format called EMF-UML. The model transformations were realized using ATL[8] for model-to-model (M2M) and JET[9] for model-to-text (M2T) transformations. Both Eclipse-based technologies were combined in a transformation chain that is illustrated in Fig. 2.

The process starts with a UML source model (with applied UWE profile) that contains both the platform-independent model (PIM) and individual element mappings of the concrete presentation model. A first model-to-model transformation converts it to an instance of the UWE metamodel which is then validated using a set of OCL constraints. If the validation succeeds, a next transformation generates a platform-specific model (PSM) by processing the UWE source model together with an additional input model containing the default UI element configurations of the concrete presentation model. This PSM is finally used as input for a model-to-text transformation that generates the application's source code which consists of Java classes, page specifications and configuration files. These generated artefacts build upon an intermediate platform, called the UWE4JSF

[6] http://www.pst.ifi.lmu.de/projekte/uwe/uwe4jsf
[7] Eclipse Modeling Framework — http://www.eclipse.org/modeling/emf
[8] Atlas Transformation Language — http://www.eclipse.org/m2m/atl/
[9] http://www.eclipse.org/modeling/m2t

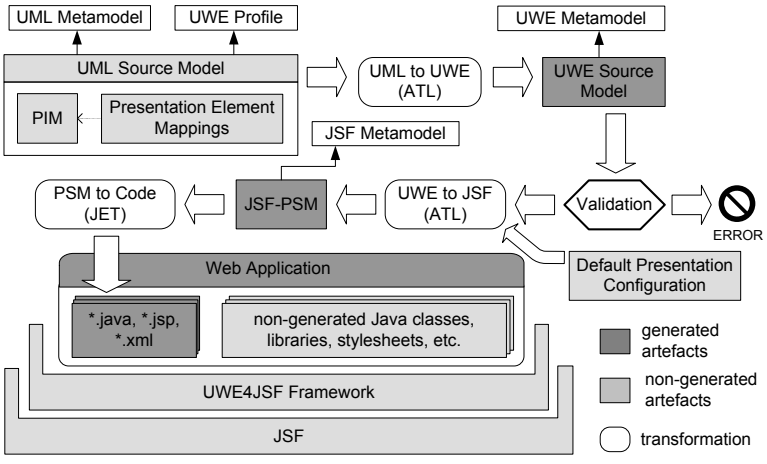

Fig. 2. UWE4JSF Generation Process

framework, that resides on top of JSF and is designed to reduce the complexity of the generated code and the transformation rules. Finally, the application can be augmented with non-generated Java classes to implement complex process actions or persistence layer operations.

4 Conclusions and Future Work

We presented the MDD tool UWE4JSF that allows fully automatic generation of JSF-based web applications from UWE models. The chosen platform provides high extensibility and UWE's concrete presentation model can be used to exploit the vast amount of available JSF component libraries that provide means to create rich user interfaces. JSF also offers a flexible architecture for supporting simultaneously multiple platforms like browsers technologies and mobile devices. Future work includes efforts to incorporate mechanisms like pattern substitution and to extending the validation engine. Last but not least, the MagicUWE[10] project aims to create an elaborate integration of UWE/UWE4JSF into a professional UML CASE tool.

References

1. Kraus, A., Knapp, A., Koch, N.: Model-Driven Generation of Web Applications in UWE. In: MDWE2007 - 3rd International Workshop on Model-Driven Web Engineering, July 2007. CEUR-WS, vol. 261, pp. 23–38 (2007)
2. Kroiss, C.: Model-based Generation of Web Applications with UWE (in German). Diploma Thesis. Ludwig-Maximilians-Universität München, Germany (June 2008), `http://www.pst.ifi.lmu.de/projekte/uwe/publications/CKroissDA08.pdf`

[10] http://www.pst.ifi.lmu.de/projekte/uwe/toolMagicUWE.html