

Modelling the Behaviour of Web Applications with ArgoUWE*

Alexander Knapp¹, Nora Koch^{1,2}, and Gefei Zhang¹

¹ Ludwig-Maximilians-Universität München
{knapp, kochn, zhangg}@pst.ifi.lmu.de
² F.A.S.T. GmbH, Germany
koch@fast.de

Abstract. A methodology needs to be empowered by appropriate tool support. The CASE tool ArgoUWE supports designers in the use of the UWE methodology for the systematic, UML-based development of Web applications. ArgoUWE is implemented as a plugin of the open source ArgoUML modelling tool. Besides extending ArgoUML by features for modelling conceptual, navigational, and presentation structures, the new version of ArgoUWE includes support for modelling the behavioural aspects of workflow-driven Web applications. Moreover, ArgoUML's design critic mechanism has been extended to give continuous feedback of deficiencies and inconsistencies in UWE models during the modelling process.

Web applications have evolved from Web information systems to workflow-driven software systems supporting business processes. Therefore, models of Web applications had evolved, too, to explicitly support the business process view. The UWE methodology [6] allows for the specification of conceptual, navigation, and presentation models for Web applications following the clear separation of the three structural Web-inherent aspects: content, hypertext, and layout. Additionally, UWE integrates a business process view by offering models for specifying the process structure and behaviour of business workflows, such as those used for the design of reservation systems (e.g., for flights or hotels) or e-commerce applications such as online book shops, and combining processes into the conceptual, navigation, and presentation structure.

We present a new version of the ArgoUWE [4, 5] CASE tool¹ which includes modelling facilities to support the UWE-based design of process-driven Web applications. A business process model and new stereotyped classes are included as well as a set of additional constraints that are used for consistency checks and semi-automatic generation of process models. In contrast to older versions of ArgoUWE, consistency of models is now checked in the background during modelling. This way the developer is supported but not constrained in his modelling activities with suggestions for corrections and improvements.

* This research has been partially sponsored by the EC 5th Framework project AGILE (IST-2001-32747) and Deutsche Forschungsgemeinschaft (DFG) within the project MAEWA (WI 841/7-1)

¹ <http://www.pst.ifi.lmu.de/projekte/argouwe>

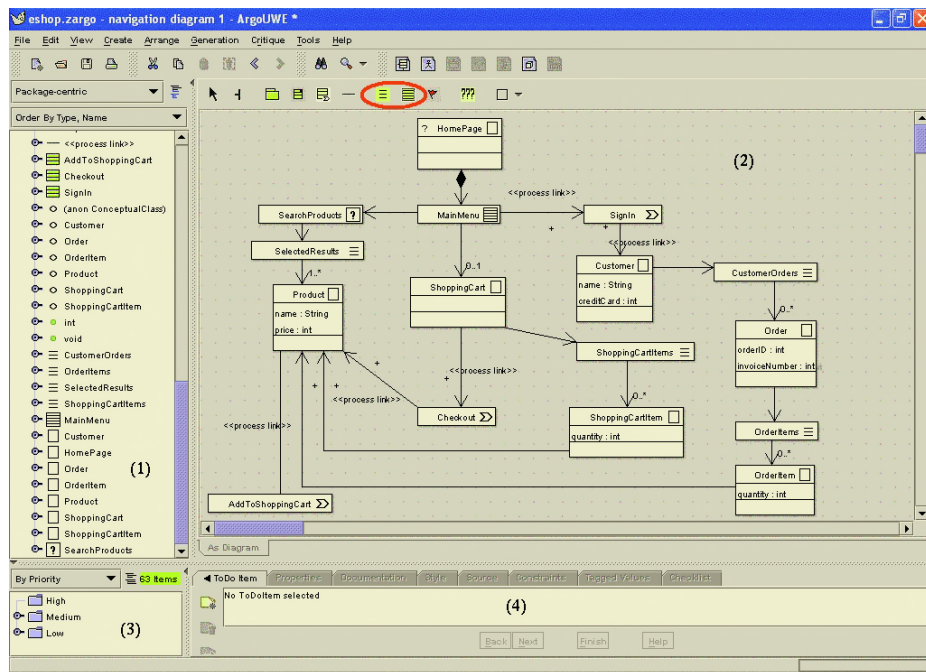


Fig. 1. Navigation model after integration of process nodes

In ArgoUWE, business process modelling is integrated in the design on the levels of use case modelling, process structure and process flow modelling, and navigation modelling. On the use case level, separation of `<<navigational>>` use cases and process use cases is offered. Both kinds of use cases are combined into a navigation structure model (see Fig.1): ArgoUWE generates a `<<process node>>` (visualised by an arrow symbol) in the navigation model for each (non-navigational) use case that is manually selected by the modeller. Thereby, a `<<process class>>` is generated for the `<<process node>>` of the selected use case and automatically included in the process structure model. A process structure model is represented by a UML class diagram and describes the relationship of a `<<process node>>` and other `<<process class>>`es whose instances are used to support this business process. The logic of the business process is described by a process flow model visualised as a UML activity diagram.

ArgoUWE is implemented as a plugin of the open-source UML-CASE tool ArgoUML². ArgoUWE makes use of ArgoUML's general graphical user interface (see Fig. 1) and thus is intuitive to ArgoUML users. In particular, the *explorer pane* (1) provides a tree structure of all diagrams and model elements of the current project. A single UWE diagram is edited in the *editing pane* (2). Issues of design critiques are listed in the *to-do pane* (3), sorted by several possible criteria like priority or the model element causing the critique. The *details pane* (4) comprises several panels for showing and editing details of the currently selected model element or to-do item.

² <http://www.argouml.org>

On the one hand, ArgoUWE provides the necessary diagram types for modelling Web-applications with the UWE method: conceptual, navigation, and presentation diagrams, as well as process structure and process flow. On the other hand, it also supports semi-automatic model transformations as, for example, from the conceptual to the navigation model by including access primitives (encircled icons in Fig. 1). Finally, ArgoUWE extends ArgoUML's design critic mechanism for providing continuous feedback on modelling deficiencies and inconsistencies based on the UWE metamodel and its OCL constraints.

Only few Web engineering methods support the systematic development of Web applications with a mature CASE tool. The most advanced tool support is offered for the method OO-H [3] and the modelling language WebML [1] with their CASE tools VisualWADE and WebRatio, respectively. VisualWADE provides an operational environment that in contrast to ArgoUWE uses the UML only in the first phase of the development process but has the advantage to allow the designers to render the final look and feel of the application. WebRatio supports modelling of Web applications using the ER notation and the proprietary Web Modelling Language WebML differing from UWE as it does not perform a clear separation of the navigation and presentation aspects. A more architecture-oriented notation is proposed by Conallen [2], which is supported by the IBM Rational ModellerTM tool, but in contrast to ArgoUWE it neither supports a systematic development process nor guides the developer through the Web-specific process.

References

1. Stefano Ceri, Pietro Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, and Maristella Matera. *Designing Data-Intensive Web Applications*. Morgan-Kaufmann, San Francisco, 2002.
2. Jim Conallen. *Building Web Applications with UML*. Addison-Wesley, Reading, Mass., &c., 2nd edition, 2003.
3. Jaime Gómez, Cristina Cachero, and Oscar Pastor. On Conceptual Modeling of Device-Independent Web Applications: Towards a Web-Engineering Approach. *IEEE Multimedia*, 8(2):26–39, 2001.
4. Alexander Knapp, Nora Koch, and Gefei Zhang. Modeling the Structure of Web Applications with ArgoUWE. In Nora Koch, Piero Fraternali, and Martin Wirsing, editors, *Proc. 4th Int. Conf. Web Engineering (ICWE'04)*, volume 3140 of *Lect. Notes Comp. Sci.*, pages 615–616. Springer, Berlin, 2004.
5. Alexander Knapp, Nora Koch, Gefei Zhang, and Hanns-Martin Hassler. Modeling Business Processes in Web Applications with ArgoUWE. In Thomas Baar, Alfred Strohmeier, Ana M. D. Moreira, and Stephen J. Mellor, editors, *Proc. 7th Int. Conf. Unified Modeling Language (UML'04)*, volume 3273 of *Lect. Notes Comp. Sci.*, pages 69–83. Springer, Berlin, 2004.
6. Nora Koch and Andreas Kraus. The Expressive Power of UML-based Web Engineering. In Daniel Schwabe, Oscar Pastor, Gustavo Rossi, and Luis Olsina, editors, *Proc. 2nd Int. Wsh. Web-Oriented Software Technology (IWWOST'02)*, pages 105–119. CYTED, 2002.