

# WebRatio 5: An Eclipse-Based CASE Tool for Engineering Web Applications

Roberto Acerbis<sup>1</sup>, Aldo Bongio<sup>1</sup>, Marco Brambilla<sup>2</sup>, and Stefano Butti<sup>1</sup>

<sup>1</sup> WebModels S.r.l.

Piazzale Gerbetto, 6. I22100 Como, Italy

{roberto.acerbis, aldo.bongio, stefano.butti}@webratio.com

<sup>2</sup> Dipartimento di Elettronica e Informazione, Politecnico di Milano

Piazza L. Da Vinci, 32. I20133 Milano, Italy

mbrambil@elet.polimi.it

**Abstract.** The goal of this work is to present the software WebRatio 5, which is a good representative of a new generation of CASE tools for model-driven design of Web applications. WebRatio 5 supports the WebML language and methodology, and exploits the implementation experience of previous versions of the software for providing user-friendly application design paradigms and reliable code generation engines. The tool is developed as a set of Eclipse plugins and takes advantage of all the features of this IDE framework. Moreover, it provides new capabilities in terms of support of extensions to the models, project documentation, and coverage of new phases of the development lifecycle. The overall approach moves towards a full coverage of the specification, design, verification, and implementation of Web applications.

## 1 Introduction and Motivation

Although new paradigms of Web applications are arising, data-intensive Web applications still constitute the most diffused class of applications found on the Web. Since their size and complexity are typically high, the ideal software development process for this kind of applications should meet two goals: (i) incorporate requirements and model driven design in the development lifecycle; (ii) delivering a software architecture that meets the non-functional requirements of performance, security, scalability, availability, maintainability, usability, and high visual quality. Such process should also be amenable to automation, to let developers concentrate on functional requirements and optimization, and delegate the repetitive tasks (such as code implementation) to software tools.

The model-driven design of this kind of Web applications should start from well established requirement specifications and involves the definition of a data model (to specify the data used by the application), a hypertext model (to describe the organization of the front-end interface) and a presentation model (to personalize the graphical aspect of the interface). Afterwards, model verification and model transformations (e.g., for generating the running code) should be provided to complete the development process. Unfortunately, no existing CASE tool can claim to support all these aspects of the development.

In this work we present the CASE software WebRatio 5 [10], representing a new generation of model-driven CASE tools for Web applications. WebRatio 5 supports the WebML language and methodology, and exploits the implementation experience of previous versions of the software for providing user-friendly application design paradigms and reliable transformation engines. The tool is developed as a set of Eclipse plug-ins and takes advantage of its features. Moreover, it provides new capabilities in terms of support of extensions to the models, project documentation, and coverage of new phases of the development lifecycle.

The main advantages of the new Eclipse version of WebRatio are the following:

- All the design and development activities are performed through a common interface. This includes the modeling of the Web application, the definition of the visual identity, and the development of new business components;
- All the design items (models, components, documentation, and so on) are stored into a common area (the so-called Eclipse workspace) and can be easily versioned into a versioning system, such as CVS;
- All the existing Eclipse widgets can be reused and integrated in the toolsuite;
- New and existing editors for model and code design can be easily integrated.

The following sections outline the main features of the tool, provide some GUI examples and describe the overall philosophy of the WebRatio toolsuite, more and more moving towards a full coverage of development process of Web applications.

## 2 Supporting the Design of WebML Models

WebRatio 5 fully supports the WebML metamodel [9] [3], including the most recent extensions for workflow-driven Web applications [2] and Web services [8]. WebML is a high-level notation for data-, service-, and process- centric Web applications. It allows specifying the data model of a Web application and one or more hypertext models (e.g., for different types of users) used to publish and manipulate the underlying data. Each hypertext is a graph of *pages*, consisting of connected *units*, representing at a conceptual level the primitives for publishing contents into pages. Units are connected by *links*, that define navigation paths and carry data to allow computation of the hypertext. Hypertexts also include *operations* that specify business actions, such as content management operations on the data or other kinds of tasks.

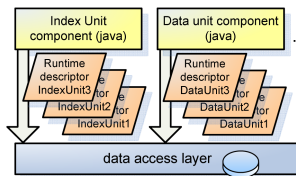


Fig. 1. Java components and XML descriptors for WebRatio units

### 3 WebRatio 5 Architecture

The *design-time* part of WebRatio 5 is a GUI for application design comprising a set of editors, a set of transformers and model validators, and some workspace management components. Models are saved as XML documents. The model is transformed into a running Web application through a code generator, which is developed using the ANT, XSLT, and Groovy technologies. Groovy [7] is an agile language using a Java-like syntax and fully integrated in the Java Platform. It provides many features that are inspired by scripting languages. The tool is integrated with CVS [1] for collaborative design and visual synchronization of project versions. The architecture is fully extensible, since it allows to specify new components (units) and include them in the application design and code generation framework. At design time, the components are described through a Java class that implements the service of the component and by a set of XML descriptor, defining its interface.

The *run-time* framework exploits a set of off-the-shelf object-oriented components for organizing the business tier:

- *Smart service creation*: services that implement units or business actions are created upon request, cached, and reused across multiple requesters;
- *Activity log*: a set of pre-built functions for logging each service is provided;
- *XML parsing and access*: access to the information stored in the XML unit descriptors is granted by standard parsing tools;
- *Connection pooling*: pre-built functions for dynamically managing a pool of database connection allow to optimize performance and reliability.

At runtime one single service class is deployed for each type of component (which is then instantiated with the smart service creation approach). Moreover, one runtime XML descriptor is deployed for each component used in the design (Fig. 1).

### 4 WebRatio 5 GUI

WebRatio 5 has been implemented as a set of Eclipse [5] plug-ins. Eclipse is a framework for IDEs, in the sense that, besides being an IDE itself, it provides the infrastructure for defining new IDEs, i.e., new plug-ins for a particular programming language or model. For instance, plug-ins exist for Java, C/C++, Python, Perl, UML, and many others. Eclipse is an open source multi-platform framework, executable on Linux, Windows, and Mac OS X.

The WebRatio GUI defines a special Eclipse perspective designed to better suit the needs of visual modelling. It comprises several panels, which include:

- *Model diagram editors* for the design of the WebML data model and hypertext models. The diagram editors are based on the GEF [6] framework and libraries. GEF is a very powerful framework for visually creating and editing models. Fig. 2 (a) shows a snapshot of the data model editor, comprising the project tree, the main diagram editor, the component panel (bottom left), the property panel (center), and the error panel (bottom right);

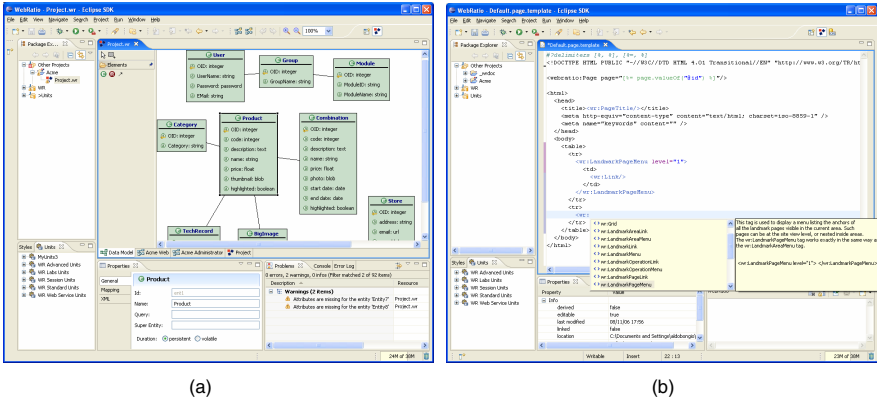


Fig. 2. Data model editor and HTML editor (with code completion) in WebRatio 5

- *Advanced text editors* for the design of XML descriptors, Java components, and so on. The editors provide typical features like syntax highlighting, auto-completion, and so on. An example is shown in Fig. 2 (b);
- *Form-based editors* for the specification of new components and for the properties of components instances. Fig. 3 (a) shows the editor of Indexes;
- *Wizards* for the support of the most common design tasks (e.g., new projects);
- *Documentation editors* for refined and customized project documentation generation. For instance, Fig. 3 (b) shows the editing and the generated documentation for the Index unit component.

The long-term focus of the WebRatio 5 is oriented towards the full coverage of the development process. In this sense, some new *beta* pieces are being developed, for project documentation generation and coverage of new design steps. For instance, a new fully-integrated Business Process editor and WebML model generator [1] is now available (Fig. 4 shows a sample snapshot). It generates skeletons of WebML models that comply with the BP specification and can be refined by the designer.

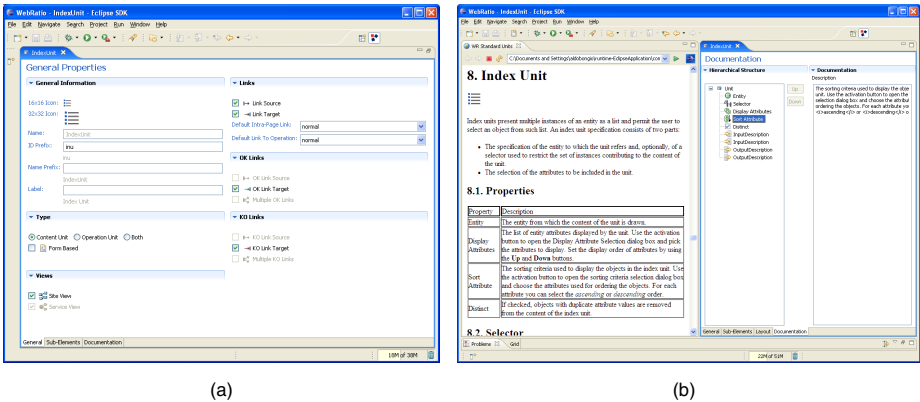


Fig. 3. Component definition editor and project documentation editor in WebRatio 5

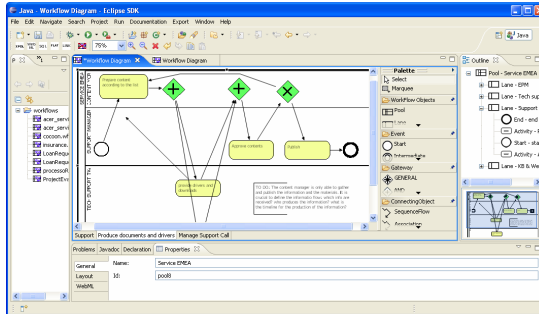


Fig. 4. The BPMN business process editor for Eclipse

## 5 Conclusions

The proposed demonstration illustrates WebRatio 5, a CASE tool based on the Eclipse framework that allows the model-driven specification of a complex Web application, including process management primitives, calls to Web services, and integration of heterogeneous data sources. The demonstration shows that application developers can concentrate only on the requirements of the application and on its high-level design, because code and project documentation are automatically generated by the CASE tool and correctness is automatically verified.

## References

- [1] Brambilla, M.: Generation of WebML Web Application Models from Business Process Specifications. Demo at ICWE2006, pp. 85–86. ACM Press, New York (2006)
- [2] Brambilla, M., Ceri, S., Fraternali, P., Manolescu, I.: Process Modeling in Web Applications. In: ACM TOSEM, vol. 15(4), pp. 360–409 (2006)
- [3] Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: Designing Data-Intensive Web Applications. Morgan Kaufmann, San Francisco (2002)
- [4] CVS, Concurrent Versions System (2007) <http://www.nongnu.org/cvs/>
- [5] Eclipse (2007) <http://www.eclipse.org/>
- [6] Eclipse GEF (2007) <http://www.eclipse.org/gef/>
- [7] Groovy (2007) <http://groovy.codehaus.org/>
- [8] Manolescu, I., Brambilla, M., Ceri, S., Comai, S., Fraternali, P.: Model-Driven Design and Deployment of Service-Enabled Web Applications. ACM TOIT 5(3), 439–479 (2005)
- [9] WebML.org. (2007) <http://www.webml.org>
- [10] WebRatio 4.3. (2007) <http://www.webratio.com/>