

# Model-Driven Development Based on OMG's IFML with WebRatio Web and Mobile Platform

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

<sup>1</sup> WebRatio Srl., Como, Italy  
{roberto.acerbis,aldo.bongio,stefano.butti}@webratio.com,  
marco.brambilla@polimi.it

<sup>2</sup> Politecnico di Milano. Dip. di Elettronica, Informazione e Bioingegneria,  
Milan, Italy


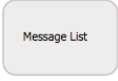




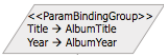
**Abstract.** The need for quick development of mobile applications, combined with the chronic shortage of IT personnel with skills in Web and mobile app development advocate for new approaches and tools. In this context, we propose a comprehensive tool suite called WebRatio Platform for model-driven development of web and mobile applications. The tool supports developers in the specification of the domain model and of the user interaction model for applications according to two extended versions of the OMG standard language called IFML (Interaction Flow Modeling Language). The extensions feature primitives tailored to web application development and to mobile systems. The tool features model-checking and full code generation that produces ready-to-publish Web and cross-platform mobile applications.

## 1 Introduction

Several researches have applied software engineering and Web engineering techniques to the specification of Web and multi-platform application interfaces and user interaction in broad sense. Among them, we can cite OO-HDM [8], WAE [2], WebDSL [4], OOH-Method [3], WebML [1], RUX-Model [6], HERA [9], and rapid UI development [7] and modeling languages like USIXML [5]. Commercial vendors are nowadays proposing tools for Web development, like Mendix (<http://www.mendix.com>) and Outsystems (<http://www.outsystems.com>) However, none of them has managed to become widely adopted in the software industry yet. For this reason, front-end development continues to be a costly and inefficient process, where manual coding is the predominant development approach, reuse is low, and cross-platform portability remains difficult.

In 2014, the Object Management Group has adopted a new standard called Interaction Flow Modeling Language (IFML), which supports the platform independent description of graphical user interfaces for applications accessed or deployed on such systems as desktop computers, laptop computers, PDAs, mobile phones, and tablets. IFML is inspired by WebML but adds several innovations: it increases separation of concerns, completely forbidding the integration of business logic into the user interaction specification; it defines a set of very generic concepts (the core of the language) which can be applied to any kind of

**Table 1.** Basic Core IFML concepts: description, notation, and platform-specific cases

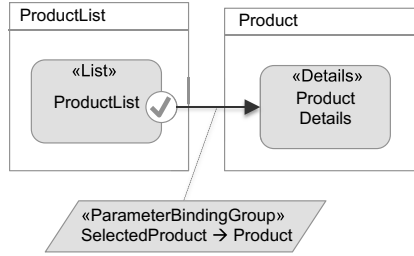
| Concept                 | Meaning  | IFML Notation   | PSM Example  |
|-------------------------|--|---|--|
| View Container          | An element of the interface that comprises elements displaying content and supporting interaction and/or other ViewContainers. |  | Web page<br>Window<br>Pane.                                    |
| View Component          | An element of the interface that displays content or accepts input   |  | An HTML list.<br>A JavaScript image gallery.<br>An input form. |
| Event                   | An occurrence that affects the state of the application  |  |  |
| Action                  | A piece of business logic triggered by an event  |  | A database update.<br>The sending of an email.                 |
| Navigation Flow         | An input-output dependency. The source of the link has some output that is associated with the input of the target of the link |  | Sending and receiving of parameters in the HTTP request        |
| Data Flow               | Data passing between ViewComponents or Action as consequence of a previous user interaction.                                   |  |  |
| Parameter Binding Group | Set of ParameterBindings associated to an InteractionFlow (being it navigation or data flow)                                   |  |  |

user interface; it brings in the concept of event and asynchronous interactions; and it integrates seamlessly with UML and BPMN notations.

In this paper we describe our experience in implementing IFML for the Web and mobile domains [4] within the tool called WebRatio Platform, a model-driven, low-code development platform based on Eclipse that generates fully-functional Web applications and cross-platform mobile applications based on the Apache Cordova / PhoneGap framework.

## 2 The Interaction Flow Modeling Language (IFML)

The Interaction Flow Modeling Language (IFML) supports the platform independent description of graphical user interfaces for applications accessed or deployed on any systems and technological platform. The focus of the description is on the structure and behavior of the application front-end as perceived by the end user. With respect to the popular Model-View-Controller (MVC) model of an interactive application, the focus of IFML is mainly on the view part. IFML models support the following design perspectives (whose notation is summarized in Table 1): (1) The *view structure specification*, which consists of the definition of view containers; (2) The *view content specification*, which consists of the definition of view components, i.e., content publishing and data



**Fig. 1.** Example of IFML model showing a list of products and the details view

entry elements contained within view containers; (3) The *events specification*, which consists of the definition of events (coming from users interaction, application logic, or external agents) that may affect the state of the UI; (4) The *event transition specification*, which consists of the definition of the effect of an event on the user interface; and (5) The *parameter binding specification*, which consists of the definition of the input-output dependencies between model elements. Furthermore, IFML can be complemented with external models for connecting to any kind of content model (representing databases, ontologies, file systems or other resources) and any kind of dynamic model (describing the business logic behind the application front end).

Figure 1 shows a simple example of IFML diagram, where a starting page displays a list of products and, upon selection by the user, a target page shows the details of the selected product.

### 3 WebRatio Platform

WebRatio Platform ([www.webratio.com](http://www.webratio.com)) is a model-driven low-code development tool based on IFML, which features two editions, respectively focusing on the design and development of Web and mobile applications. WebRatio Platform is implementing the Web-extended version of IFML and focusing specifically on Web applications. WebRatio Mobile Platform is implementing the mobile-extended version of IFML and focusing specifically on mobile applications. WebRatio provides three integrated environments:

- The modeling environment supporting the specification of IFML diagrams for the view description, UML class diagrams (or ER diagrams) for the information design, and optionally the integration with BPMN diagrams for the specification of business process aspects.
- The development environment for supporting the implementation of custom components, at the purpose of allowing personalized extensions of the modeling language based on vertical needs or native functionalities, as well as custom data and system integration requirements
- The layout template and style design environment, which allows the highest possible level of UI sophistication, thanks to full support of HTML 5, CSS and JavaScript based styling.

Based on the input provided through these environments, WebRatio provides model checking, full code generation, group-work support and lifecycle management. The generated code consists of: automatic cloud-deployed Java EE code covering both front-end of back-end of web applications for the Web version of WebRatio; and ready-to-deploy cross-platform mobile applications, based on the open-source Apache Cordova framework, within the PhoneGap distribution, for the Mobile version of WebRatio. The layout and styling of the target apps can be defined through the definition of HTML5 templates and CSS. By default, WebRatio Mobile builds apps that are fully functional even when not connected to the Internet (offline mode). It also provides a built-in solution for read/write data synchronization between the mobile device and the back end. The mobile app architecture is based on open and de facto standard libraries, such as Angular.js, JData, and Ionic.

In the deploy, integration and coherency between mobile and web application is granted by a common modeling approach. Web and mobile applications can be integrated with any existing back-end system with RESTful APIs; alternatively, one can build his own back-end using WebRatio Platform and a time-based cloud plans. The generated code base is human readable and maintainable, even outside WebRatio Platform. There are no proprietary or closed components in the generated code.

## References

1. Ceri, S., Brambilla, M., Fraternali, P.: The history of WebML lessons learned from 10 years of model-driven development of web applications. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) *Conceptual Modeling: Foundations and Applications*. LNCS, vol. 5600, pp. 273–292. Springer, Heidelberg (2009)
2. Conallen, J.: *Building Web applications with UML*. Addison Wesley (2002)
3. Gómez, J., Cachero, C., Pastor, O.: *Conceptual modeling of device-independent web applications*, pp. 26–39 (2001)
4. Groenewegen, D.M., Hemel, Z., Kats, L.C.L., Visser, E.: WebDSL: a domain-specific language for dynamic web applications. In: Harris, G.E. (ed.) *OOPSLA Companion*, pp. 779–780. ACM (2008)
5. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., López-Jaquero, V.: USIXML: a language supporting multi-path development of user interfaces. In: Feige, U., Roth, J. (eds.) *DSV-IS 2004 and EHCI 2004*. LNCS, vol. 3425, pp. 200–220. Springer, Heidelberg (2005)
6. Linaje, M., Preciado, J.C., Sánchez-Figueroa, F.: A method for model based design of rich internet application interactive user interfaces. In: Baresi, L., Fraternali, P., Houben, G.-J. (eds.) *ICWE 2007*. LNCS, vol. 4607, pp. 226–241. Springer, Heidelberg (2007)
7. Schramm, A., Preußner, A., Heinrich, M., Vogel, L.: Rapid UI development for enterprise applications: combining manual and model-driven techniques. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) *MODELS 2010, Part I*. LNCS, vol. 6394, pp. 271–285. Springer, Heidelberg (2010)
8. Schwabe, D., Rossi, G., Barbosa, S.D.J.: Systematic hypermedia application design with OOHDM. In: *Proc. Hypertext 1996*, pp. 116–128 (1996)
9. Vdovják, R., Fräsincar, F., Houben, G.-J., Barna, P.: Engineering Semantic Web Information Systems in Hera. *Journal of Web Engineering* 1(1–2), 3–26 (2003)