



Web Modeling Languages Strike Back

Gustavo Rossi • *Universidad Nacional de La Plata, Argentina*

Some weeks ago, the Object Management Group (OMG; www.omg.org) – an international, open membership, not-for-profit computer industry standards consortium – adopted a new standard, the Interaction Flow Modeling Language (IFML; www.ifml.org). In the past, OMG has adopted many influential languages and notations, including UML and the Business Process Model and Notation (BPMN) language. As described on its site, IFML is designed to express the content, user interaction, and control behavior of the front end of applications belonging to domains such as traditional HTML+HTTP-based Web applications; rich Internet applications, as supported by the forthcoming HTML 5 standard; mobile applications; and multichannel and context-aware applications.

IFML is an interesting new resource for Web developers; my aim here isn't to fully describe the standard but to briefly explain why its emergence positively affects us, and what we can expect in the near future (although talking about the future is never easy when it comes to the Internet). For the complete specification and meaningful examples, interested readers can refer to current documents on the IFML site.

Web Applications and Modeling

The Web is the most important face of the Internet and has evolved from a platform for publishing static material to the preferred vehicle for learning, business, and entertainment. Users are no longer passive readers of Web content but rather provide part of this content and even mash it up with services to create new applications. New software development paradigms such as service-oriented and cloud computing are based on Web technologies. This fast evolution explains, at least in part, that despite the myriad new development environments, building novel

Web applications is harder even when using tools for composing content and services.

For (Web) software engineers, this is a well-known problem, given that facing software change is always a nightmare, and the Web changes constantly: everyday new possibilities arise that trigger new requirements, and, in contrast with the “old” pre-Web software, users are more than aware of these possibilities. A wonderful summary of the problems of engineering Web software is available elsewhere.¹

The smartest way to deal with Web software evolution's different facets is to use model-driven approaches – that is, to raise the level of abstraction in which we think about Web applications by using modeling rather than low-level programming languages, and derive programs “automatically” via model transformations. Using models to build Web applications gives us some additional advantages because we can describe complex functionality (such as rich interaction features) without delving into implementation details. Deriving complex Web applications directly from models is now feasible, practical, and without performance loss; some authors have even derived design models from existing applications,² thus improving the process of reengineering legacy Web software.

In this context, Web modeling languages have played a major role in the past decade. Since the late nineties, it's become clear that existing languages (such as UML) fall short when it comes to expressing the specific and new types of behavior typical of Web applications, such as navigation, page composition, and interactive behaviors. Thus, many modeling and methodological approaches emerged, and for the past 10 years, developers willing to use model-driven approaches had to evaluate which one to choose. All these approaches adopted the “old” modeling languages (entity/relationships, UML, and

so on) for expressing general application features, but the application's navigation structure and related behavior were expressed with brand new notations. An interesting case regarding the disjunctive between using standard or developing new notations has been UML-based Web Engineering (UWE; <http://uwe.pst.fh-lmu.de/index.html>), wherein its developers managed to extend and adapt the UML standard for the Web realm. I'll look at this use of standards later. Interested readers can find a thorough presentation of the most mature Web design approaches elsewhere,³ all of them applied to solve the same problem for the sake of comparison.

Development of a Standard

The emergence of a standard such as IFML not only shows that the

field has matured; considering the fact that several companies have worked together to develop the standard, it clearly indicates the software industry's interest in supporting this style of Web software development.

IFML is strongly inspired by a Web modeling approach, the Web Modeling Language (WebML), originally proposed by researchers at Politecnico di Milano, Italy (www.polimi.it).⁴ WebML is supported by a software tool, WebRatio (maintained by one of the companies that pushed for the new standard; see www.webratio.com). Similarly to other modeling approaches, WebRatio enables developers to express webpages' structure, the information entities from which they take their content, and the hypertext relationships among pages.

IFML is now strictly bound to UML; quoting the IFML definitions from its website,

An IFML diagram consists of one or more top-level *view containers*, representing UI windows or Web pages. A view container can contain *view components*, which denote the publication of content or interface elements for data entry (for example, input forms). A view component can have input and output parameters. A view container and a view component can be associated with events, to denote that they support the user's interaction.

In IFML diagrams, developers can add references to classes, methods, and other UML artifacts. IFML isn't necessarily "better" than previous approaches; for example, UWE could have been "the" standard, and, in fact,



Editor in Chief

Michael Rabinovich • michael.rabinovich@case.edu

Associate Editors in Chief

M. Brian Blake • m.brian.blake@miami.edu
Barry Leiba • barryleiba@computer.org
Maarten van Steen • steen@cs.vu.nl

Editorial Board

Virgilio Almeida • virgilio@dcc.ufmg.br
Elisa Bertino • bertino@cerias.purdue.edu
Fabian Bustamante • fabianb@cs.northwestern.edu
Vinton G. Cerf • vint@google.com
Siobhán Clarke • siobhan.clarke@cs.tcd.ie
Fred Dougliis* • f.dougliis@computer.org
Schahram Dustdar • dustdar@dsg.tuwien.ac.at
Stephen Farrell • stephen.farrell@cs.tcd.ie
Robert E. Filman* • filman@computer.org
Carole Goble • cag@cs.man.ac.uk
Michael N. Huhns • huhns@sc.edu
Anne-Marie Kermarrec • anne-marie.kermarrec@inria.fr
Samuel Madden • madden@csail.mit.edu
Anirban Mahanti • anirban.mahanti@nicta.com.au
Cecilia Mascolo • cecilia.mascolo@cl.cam.ac.uk
Pankaj Mehra • pankaj.mehra@ieee.org
Peter Mika • pmika@yahoo-inc.com
Dejan Milojčić • dejan@hpl.hp.com
George Pallis • gpallis@cs.ucy.ac.cy
Charles J. Petrie* • petrie@stanford.edu
Gustavo Rossi • gustavo@lifia.info.unlp.edu.ar
Amit Sheth • amit.sheth@wright.edu
Munindar P. Singh* • singh@ncsu.edu

Oliver Spatscheck • oliver@spatscheck.com
Torsten Suel • suel@poly.edu
Craig W. Thompson • cwt@uark.edu
Doug Tygar • tygar@cs.berkeley.edu
Steve Vinoski • vinoski@ieee.org
* EIC emeritus

CS Magazine Operations Committee

Paolo Montuschi (chair), Erik R. Altman,
Nigel Davies, Lars Heide, Simon Liu, Cecilia Metra,
Shari Lawrence Pfleger, Michael Rabinovich,
Forrest Shull, John R. Smith, Gabriel Taubin,
George K. Thiruvathukal, Ron Vetter, and Daniel Zeng

CS Publications Board

Thomas M. Conte (chair), Alain April, David Bader,
Angela R. Burgess, Greg Byrd, Koen DeBosschere,
Frank E. Ferrante, Paolo Montuschi, Linda I. Shafer,
and Per Stenström

Staff

Editorial Management: Rebecca Deuel-Gallegos
Lead Editor: Linda World, lworld@computer.org
Publications Coordinator: internet@computer.org
Contributors: Rebecca Olgeirson, Keri Schreiner,
and Joan Taylor

Director, Products & Services: Evan Butterfield
Senior Manager, Editorial Services: Robin Baldwin
Manager, New Media & Production: Steve Woods
Senior Business Development Manager: Sandy Brown
Membership Development Manager: Cecelia Huffman
Senior Advertising Supervisor: Marian Anderson,
manderson@computer.org

Technical cosponsor:



IEEE Internet Computing
IEEE Computer Society Publications Office
10662 Los Vaqueros Circle
Los Alamitos, CA 90720 USA

Editorial. Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Internet Computing* does not necessarily constitute endorsement by IEEE or the IEEE Computer Society. All submissions are subject to editing for style, clarity, and length.

Submissions. For detailed instructions, see the author guidelines (www.computer.org/internet/author.htm) or log onto *IEEE Internet Computing's* author center at ScholarOne (<https://mc.manuscriptcentral.com/cs-ieee>). Articles are peer reviewed for technical merit.

Letters to the Editors. Email lead editor Linda World, lworld@computer.org

On the Web. www.computer.org/internet/

Subscribe. Visit www.computer.org/subscribe/.

Subscription Change of Address. Send requests to address.change@ieee.org.

Missing or Damaged Copies. Contact help@computer.org.

To Order Article Reprints. Email internet@computer.org or fax +1 714 821 4010.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

the two have a common grounding in UML. Although IFML notation is very expressive and based on solid semantics, the IFML team also made the strategic decision to make its notation a standard, and managed to complete all the hard work needed to establish this standard. A part of this work, as you can see on the IFML page, is technical (completing the MetaObject Facility metamodel of IFML, the XML Metadata Interchange exchange format, the UML profile specification, and so on). However, a lot of less “enjoyable” work is involved, such as defending the team’s ideas against committees, providing examples, and arguing about the need for a new standard. This work goes quite beyond pure research and needs commitment. It isn’t necessary to stress standards’

UML (perhaps the most widely known software standard) because it lacks this and that. However, even in this case, we must accept that its mere existence is a symptom of progress, a ground on which we’ll build new ideas together, collectively and cooperatively, instead of relying only on individuals.

Back to IFML, we must congratulate the standards’ promoters (see the IFML page for details) for their very smart and also hard work. For them, this is the end of a difficult road, but for the rest of us, this is the beginning of another road on which we can work together. Many will presumably not use the standard, and we can even expect new notations to emerge.

outside the standardization process.³ The new IFML, for example, recognizes that “it does not cover the modeling of the presentation issues (for example, layout, style, and look&feel) of an application front end” (see www.ifml.org). Thus, we already have a cue about how those creative individuals working on modeling issues can contribute to the standard’s progress.

Surely, brighter ideas will emerge from the IFML crew, and we will definitively read more precise (and even critical) material on the new standard in future issues of *IEEE Internet Computing*. □

Acknowledgments

I thank Marco Brambilla, one of the IFML developers, for his useful comments on this column.

References

1. S. Casteleyn et al., *Engineering Web Applications*, Data-Centric Systems and Applications, Springer, 2009.
2. M.L. Bernardi, M. Cimitile, and D. Distanto, “Web Applications Design Recovery and Evolution with RE-UWA,” *J. Software: Evolution and Process*, Sept. 2012; doi: 10.1002/smr.1561.
3. G. Rossi et al., eds., *Web Engineering: Modeling and Implementing Web Applications*, Springer, 2008.
4. S. Ceri et al., *Designing Data-Intensive Web Applications*, Morgan Kaufmann, 2002.

Gustavo Rossi is a professor at Facultad de Informática, Universidad Nacional de La Plata, Argentina, and the director of the Research and Development in Advanced IT Lab (LIFIA). He is also a researcher at CONICET. His research interests include agile approaches for Web applications development and model-driven development. Rossi has a PhD in computer science from PUC-Rio, Brazil. He’s a member of ACM and IEEE. Contact him at gustavo@lifia.info.unlp.edu.ar.

It isn’t necessary to stress standards’ importance; the Internet has been possible because different communities agreed on some basic protocol standards.

importance; the Internet itself has been possible because different communities agreed on some basic protocol standards. Later, the Web became a reality for the same reason. It would be repetitive to enumerate the large number of standards on which we daily base our work. Many times, we even ignore that they are standards, and that getting to them was a painful process usually performed by volunteers.

Unfortunately, the software development field isn’t often one in which we can find the same adherence to standards; we usually think of ourselves as more “creative” than “routine” people. Many of us criticize

However, even this is a positive for IFML and consequently for the whole community. Standards like the well-known UML usually have extension mechanisms that might be included in the language itself (for instance, extending the language metamodel). Specifically, IFML includes a basic extension package along with its core package; further packages can be devised for specific UI specification domains.

As mentioned, by using UML extension mechanisms, UWE enabled the incorporation of Web features to the former standard. Many of these features were perhaps inspired by the work of those developing approaches

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.