Towards an Interactive Web Modelling Language

J.M. Wright
Institute of Information Sciences and Technology, Massey University, Palmerston North, New Zealand

***Abstract*— Over the last few years, the increasing popularity of client-side scripting use on web sites, with rich functionality commonly known as AJAX, has been changing the face of the World Wide Web. This new standard of interactivity is quickly moving the focus from static information-based web sites to rich, interactive web applications. However, languages to model these web applications have fallen behind, and the use of software engineering approaches remains sparse in the web industry.**

This paper discusses our project to develop a standards-compliant, meta-modelled interactive web application modelling language, which we envisage will improve existing web engineering approaches by providing a current and useful model for web applications. We briefly discuss existing modelling languages and the features that they are missing, and conclude with a summary of our findings and an overview of future work.

I. Introduction

The introduction of the World Wide Web in the early 1990s allowed anybody on the Internet to browse freely-available hypertext using primitive web browsers. These early web sites were generally static pages, but the later integration of databases allowed for the first web applications to develop. Shortly thereafter the first generation of web application modelling languages started to evolve [1], aimed at improving quality and reliability.

Recent innovations to the web have ushered in a new era of interactivity and personalisation to the web. A rich client-side scripting model combined with asynchronous background network requests and new visual capabilities – commonly referred to as AJAX [2] – created a new environment for developing interactive web applications.

Existing web application modelling languages, such as WebML [3] and UWE [4], have struggled to keep up with these new concepts, and clearly improvements need to be developed to retain software engineering concepts a vital part of web development. This paper will highlight our project to develop a web modelling language suitable for describing these interactive web applications.

This paper is organised as follows. Section II is an overview of the new challenges faced in web application engineering, along with a brief overview of our project. This is followed in Section III with a summary of concepts we propose are necessary; followed in Section IV with a review of existing approaches. We conclude with a discussion of our achievements and future work in Section V.

II. Our Project

Arguably, the increased use of AJAX has transformed the expected standard for web applications on the web. These new web application feature client-side scripting (generally through Javascript), client-side Document Object Model [5] access, and the composition of asynchronous background network requests. AJAX has also allowed the development of rich client-side interfaces once only the realm of traditional desktop software, improving interactivity and productivity, and reducing total cost of ownership (TCO) to organisations due to the web's distributed model.

One popular example of AJAX is through an auto-completable *destination address* text field in an e-mail web application. As the user enters characters into this field, the client contacts the server for addresses containing these characters, displaying a list of suggested addresses. This improves usability, potentially reduces the overall bandwidth of network communication, and improves interactivity and responsiveness.

To date, current web application engineering approaches remain insufficient compared with classical software engineering in terms of its success in modelling industry standard practices; development in general remains ad hoc [6], and there remains a great need for models and technologies to improve the quality and success rate of web applications [7]. Our project aims to fulfill some of these needs, by providing an up-to-date, useful modelling language for the web, which we envisage will both serve the classical role of software design and documentation, and also fit into the Model-Driven Architecture concept (MDA: [8]) to generate executable applications from the models themselves.

The development cycle of our project follows a traditional iterative development lifecycle, allowing us to adapt to the inherent volatility of the dynamic web. Our first major task was to develop a comprehensive set of requirements that we would expect a modelling language to support. By studying existing web applications and upcoming technologies, we identified a comprehensive set of use cases to derive what new features are available to web applications. These included features such as client-side data storage and access, natural user interaction, visual effects, real-time form validation, scheduled events, distributed functionality, and offline applications; more detail is provided in [9].

Compared to existing models which tend to focus solely on the navigation and structure of web applications, these new features add significant interactivity requirements, and it is now necessary to model the

users' clients as well. Existing languages based on replacing the entire client-side interface on each request are clearly no longer appropriate.

III. New Concepts

Since AJAX technologies are still relatively new, no existing languages are available to describe this new level of interactivity, and even a comprehensive discussion on what concepts could be used does not exist. As such, we propose the following modelling concepts to address these needs, which will only be briefly discussed in this paper.

1) Event Modelling: Events are a natural part of web applications, and we argue that these should be first-class citizens of a model. It would be desirable to unify these different sources of events together, perhaps through Event-Condition-Action (ECA) rules – event sources such as server-side events (requests, scheduled events, API requests), client-side events (DOM events, user interaction), notification mechanisms (RSS) and lifecycle events (see below) can all potentially interact with each other.

2) Browser Control: The user's browser is the primary interface for interacting with a web application, and browser features such as navigation, cookies, scripting and plugins should be natively supported by a model.

3) Lifecycles: The concept of a lifecycle in web application development can assist with developing complex component-based web applications; for example, deleting shopping carts when closing a session, or creating administrator accounts when a new application is installed.

4) Users and Security: Users are an important common concept in web applications, and as such, they should have a great deal of focus and support, along with a robust security model to describe their permissions[1].

5) Databases: New concepts such as distributed databases, client-side databases and file uploads add additional requirements to modelling languages.

6) User Interface Modelling: As web applications are now more client-side oriented, it is important to be able to describe these user interfaces, and the effect of their interactions on the application state.

7) Standards/Meta-models: An important non-functional requirement is our emphasis on standards compliance. A language should be platform-independent, and should allow integration with UML [10]. Any conceptual modelling language should have MDA support, which promises tangible benefits for the quality of software engineering. It provides models with a meta-model[2] which makes model transformations[3] easier; simplifies model refactoring, transportation and code generation; eases CASE tool integration through XMI [13]; and allows rapid development of CASE tools using the EMF framework [14].

8) Verification: Software verification is an important aspect of software engineering [15], yet has little attention in web application development, other than the analysis of static navigation [16].

9) Software Support: Finally, support through software is vital for a language to be accepted by industry at large, not only as a proof-of-concept but as a reference implementation.

IV. Existing Languages

After developing these concepts, our next step was to research existing web application modelling languages, and evaluate their capabilities to our requirements. This fulfills the first testing and evaluation stages of our iterative development. This included research into the academic languages WebML [3], UWE [4], W2000 [17], OOWS [18], OOHDM [19], Araneus [20] and others; a detailed discussion into these languages is provided in [9].

Through this comprehensive academic language survey, we have come to understand that only WebML and UWE are likely to model interactive web applications sufficiently through extensions, although W2000 has a promising event-focused approach.

WebML provides a proprietary visual model for data-intensive web applications. It is a topic of active research, with many extensions proposed [21][22] and implemented in its CASE tool WebRatio [23]. The most urgent features WebML needs include a stronger events model, browser control, support for client-side scripting, and better standards support. Its Rich Internet Application (RIA) extension [24] adds client-side operation support, but events cannot be executed from diverse sources such as client-side element interaction.

UWE provides a web-oriented extension to UML with a focus on MDA concepts. UWE lacks an events model, control over the browser, and any client-side support. Many web-specific concepts such as messaging and sessions are ignored, but its standards focus should improve the success of any extensions.

Research is needed into investigating commercial web application frameworks as, such as OpenLaszlo [25] and the NetBeans Visual Web Editor [26]. Although these tend to be lower-level and more technology-oriented than academic visual modelling languages, they still provide a level of abstraction, and generally have stronger industry support.

V. Discussion

The common theme we have found with nearly all existing web application modelling languages is that they tend to lack support for modelling and

[1]Compared with the other concepts we have proposed, users and security represent a higher level of detail to a web application than core concerns such as event modelling. It's likely that these would end up in a separate model – perhaps implementable as extensions.

[2]Generally compatible with the Meta-Object Facility standard (MOF: [11]).

[3]Through existing MDA-compatible transformation languages such as QVT and ATL [12].

creating events, client-side detection and support, and sometimes even basic web-specific concepts such as sessions or sending e-mails. As such, we argue that any interactive web application modelling language should promote events to first-class citizens, and should have a strong focus on these web-specific concepts.

The next step in this research project is to start our second iteration of our development lifecycle. This entails refining our language requirements based on the knowledge we have already learned, analyse the extension mechanisms of WebML, UWE and W2000, and design and implement prototype extensions. This will allow us to understand if these languages can be extended to handle these new functionality requirements. This may prove challenging, as the initial design decision to ignore events may result in excessively complex extensions in order to "hack" this functionality on top.

Simultaneously, we have also begun prototyping our own modelling ideas. By developing our model in a modular way using the MOF-compliant EMF framework, and integrating proven standards wherever possible, we aim to understand the complexities in developing such an interactive modelling language. Consideration of existing core concepts from other languages is also important, and we will try to re-use these ideas if they fulfill our requirements. Knowledge gained from this work, along from our extension prototypes, will be used towards the development of our final approach.

I wish to acknowledge and thank my supervisor Jens Dietrich for his support and guidance throughout this project.

References

[1] F. Garzotto, P. Paolini, and D. Schwabe, "HDM – A Model-Based Approach to Hypertext Application Design," *ACM Trans. Inf. Syst.*, vol. 11, no. 1, pp. 1–26, 1993.

[2] J. J. Garrett, "Ajax: A New Approach to Web Applications," Tech. Rep., 2005. [Online]. Available: http://www.adaptivepath.com/publications/essays/archives/000385.php/

[3] S. Ceri, P. Fraternali, and A. Bongio, "Web Modeling Language (WebML): A Modeling Language for Designing Web Sites," in *Proceedings of the 9th international World Wide Web conference on Computer networks*. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 2000, pp. 137–157.

[4] N. Koch and A. Kraus, "The Expressive Power of UML-based Web Engineering," in *IWWOST'02*, 2002, pp. 105–119.

[5] W3C Group, "Document Object Model (DOM) Level 3 Core Specification," W3C Recommendation 07 April 2004, Tech. Rep., 2004. [Online]. Available: http://www.w3.org/TR/DOM-Level-3-Core/

[6] S. Murugesan and Y. Deshpande, "Meeting the challenges of web application development: the web engineering approach," in *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*. New York, NY, USA: ACM, 2002, pp. 687–688.

[7] E. Mendes, N. Mosley, and S. Counsell, "The Need for Web Engineering: An Introduction," in *Web Engineering*, E. Mendes and N. Mosley, Eds. Springer, 2006, pp. 1–27.

[8] Object Management Group (OMG), "Model-Driven Architecture Guide, v1.0.1," Tech. Rep., 2003. [Online]. Available: http://www.omg.org/cgi-bin/doc?omg/03-06-01

[9] J. Wright and J. Dietrich, "Survey of Existing Languages to Model Interactive Web Applications," in *Proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling (APCCM 2008)*, Wollongong, NSW, Australia, 2008.

[10] Object Management Group (OMG), "Unified Modeling Language (UML): Superstructure Specification, v2.0," Tech. Rep., 2005. [Online]. Available: http://www.omg.org/cgi-bin/doc?formal/05-07-04

[11] ——, "Meta Object Facility (MOF) Core Specification, v2.0," Tech. Rep., 2006. [Online]. Available: http://www.omg.org/cgi-bin/doc?formal/2006-01-01

[12] F. Jouault and I. Kurtev, "On the architectural alignment of ATL and QVT," in *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*. New York, NY, USA: ACM Press, 2006, pp. 1188–1195.

[13] C. H. Damm, K. M. Hansen, M. Thomsen, and M. Tyrsted, "Tool integration: Experiences and issues in using xmi and component technology," in *TOOLS '00: Proceedings of the Technology of Object-Oriented Languages and Systems (TOOLS 33)*. Washington, DC, USA: IEEE Computer Society, 2000, p. 94.

[14] A. Gerber and K. Raymond, "MOF to EMF: There and Back Again," in *eclipse '03: Proceedings of the 2003 OOPSLA Workshop on Eclipse Technology eXchange*. New York, NY, USA: ACM Press, 2003, pp. 60–64.

[15] G. J. Holzmann, "The Model Checker SPIN," *Software Engineering*, vol. 23, no. 5, pp. 279–295, 1997.

[16] C. Bellettini, A. Marchetto, and A. Trentini, "TestUml: User-Metrics Driven Web Applications Testing," in *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*. New York, NY, USA: ACM Press, 2005, pp. 1694–1698.

[17] L. Baresi, S. Colazzo, L. Mainetti, and S. Morasca, "W2000: A Modelling Notation for Complex Web Applications," in *Web Engineering*, E. Mendes and N. Mosley, Eds. Springer, 2006, pp. 335–364.

[18] O. Pastor, J. Fons, V. Pelechano, and S. Abrahão, "Conceptual Modelling of Web Applications: The OOWS Approach," in *Web Engineering*, E. Mendes and N. Mosley, Eds. Springer, 2006, pp. 277–302.

[19] G. Rossi and D. Schwabe, "Model-Based Web Application Development," in *Web Engineering*, E. Mendes and N. Mosley, Eds. Springer, 2006, pp. 303–333.

[20] P. Merialdo, P. Atzeni, and G. Mecca, "Design and Development of Data-Intensive Web Sites: The Araneus Approach," *ACM Trans. Inter. Tech.*, vol. 3, no. 1, pp. 49–92, 2003.

[21] I. Manolescu, M. Brambilla, S. Ceri, S. Comai, and P. Fraternali, "Model-driven design and deployment of service-enabled web applications," *ACM Trans. Inter. Tech.*, vol. 5, no. 3, pp. 439–479, 2005.

[22] M. Brambilla, S. Ceri, S. Comai, and C. Tziviskou, "Exception handling in workflow-driven Web applications," in *WWW '05: Proceedings of the 14th International Conference on World Wide Web*. New York, NY, USA: ACM Press, 2005, pp. 170–179.

[23] WebRatio Group, "WebRatio," 2007. [Online]. Available: http://www.webratio.com

[24] A. Bozzon, S. Comai, P. Fraternali, and G. T. Carughi, "Conceptual Modeling and Code Generation for Rich Internet Applications," in *ICWE '06: Proceedings of the 6th international conference on Web engineering*. New York, NY, USA: ACM Press, 2006, pp. 353–360.

[25] Laszlo Systems, Inc., "OpenLaszlo," 2006. [Online]. Available: http://www.openlaszlo.org/overview

[26] NetBeans Community, "Introduction to the NetBeans Visual Web Pack," 2007. [Online]. Available: http://www.netbeans.org/products/visualweb/