A REQUIREMENTS METAMODEL FOR RICH INTERNET APPLICATIONS

Esteban Robles Luna
LIFIA, Universidad Nacional de La Plata, Argentina
esteban.robles@lifia.info.unlp.edu.ar

María José Escalona
ETS Ingeniería Informática, University of Seville, Reina Mercedes S/N (Seville, Spain)
mjescalona@us.es

Gustavo Rossi
LIFIA, Universidad Nacional de La Plata, Argentina and Conicet
gustavo@lifia.info.unlp.edu.ar

Keywords: Rich Internet Applications, Web Engineering, Web Requirements.

Abstract: The evolution of the Web has motivated the development of several Web design approaches to support the systematic building of Web software. Together with the constant technological advances, these methods must be constantly improved to deal with a myriad of new feasible application features. In this paper we focus on the field of Rich Internet Applications (RIA); specifically we aim to offer a solution for the treatment of Web Requirements in RIA development. For this aim we present WebRE+, a requirements metamodel which incorporates RIA features into the modelling repertoire. We illustrate our ideas with a meaningful example of a business intelligence application.

1 INTRODUCTION

It is widely known that the Web is constantly evolving. In this evolution, Rich Internet Applications (RIA) (Duhl, 2003) represent a major breakthrough, as they allow combining the typical navigation flavour of the Web with the interface features of desktop applications. These applications allow reducing the communication between clients and servers since pages (differently from the "navigational" Web) do not need be fully reloaded with each user interaction. The emergence of a well-known set of RIA patterns (Yahoo Patterns) has additionally defined a small, though complete, vocabulary for expressing desired interaction functionalities in a software system. It is now common saying: "this should be an auto-complete field" or, "we can use hover details for showing this information". Not surprisingly applications stakeholders also use this vocabulary as part of their requirements for a new application.

However, though most Web design methods have been already extended to cover the scope of RIA (Meliá, et. al. 2008, Preciado et. al. 2007, Urbieta et.al. 2007), there is still an important gap in requirement specification of RIA functionality, since requirement specification and modelling languages do not include suitable primitives for expressing this kind of requirements. In this way, checking whether a requirement has been fully implemented becomes a subjective matter, and it is not possible to automate this process (e.g. by automatically generating tests from requirement specifications).

In this paper we analyze the new kind of requirements that occur in RIA, and how we can extend an existing approach to specify the behaviour of this kind of applications in a Model-Driven Web Engineering style (Moreno et. al 2007). Specifically, we use an enhanced version of the WebRE metamodel (Escalona & Koch, 2006) to specify RIA requirements.

The paper has two aims:

- We introduce a metamodel for supporting RIA requirements. The metamodel is implemented in a suitable UML profile to support reuse.
- We show how to derive a set of interaction tests from WebRE+ models to validate the RIA functionality.

The paper is structured as follows: Section 2 presents related works in Web requirements, Model-Driven Web Engineering and RIA. In Section 3, the extension of WebRE for RIA, its UML profile and its implementation in Enterprise Architect (Enterprise Architect) are presented. Section 4 presents how tests are derived from WebRE+ models. Section 5 shows a case of study with an example in the Business Intelligence area. Finally we present the conclusions and future research work in this project.

2 RELATED WORKS

The research of this paper is related with research in two different areas: the specification of Web Requirements in the context of Model Driven Engineering (MDE) and RIA. We analyze both areas in separate sub-sections.

2.1 Web Requirements Engineering and MDE

Web engineering is nowadays an important field in software engineering (Desphande et al. 2002). However there is an important gap in the treatment of requirements. In the first design approaches, OOHDM (Object-Oriented Hypermedia Design Model) (Rossi & Schwabe 2008), WebML (Web Modeling Language) (Ceri et al. 2000) or UWE (UML Web Engineering) (Koch 2008) the main focus was put on modelling and design issues, while the requirements phase was almost neglected as reported in (Escalona et. al 2007).

The importance of a full-fledge requirements phase is nowadays common in Web methodologies like NDT (Navigational Development Techniques) (Escalona & Aragon 2008) or OOWS (Fons et al. 2003). Additionally, some of the previous approaches started to include their own formalisms for requirement specification. For instance, OOHDM was enriched with UIDs (User Interface Diagrams) (Vilain et al. 2000) or WebML with business models (Brambilla et al. 2009).

Other relevant problem in the requirements specification stage is the lack of standards and therefore the proliferation of proprietary notations; each approach tends to offer its own notation. To make matters worse similar formalisms are used in different approaches with slightly different semantics, or several names are used for the same concept.

In order to solve this problem, some authors have used the concepts in MDE (Atkinson & Kühne 2003). In this development approach, building models is the main activity, and software is built by a series of model transformations ending, eventually, in a running application. Models are built using instances of concepts and relationships which are formally described by metamodels.

In (Escalona & Koch 2006) the authors present WebRE, a metamodel which defines a set of concepts to deal with requirements in Web Engineering. WebRE covers each artefact included in most methodologies such as UWE requirements, UIDs of OOHDM, W2000 extended use cases and requirements in NDT.

WebRE abstracts the concepts and terminology used in each approach and defines a common representation using shared requirements metamodel; in this way WebRE allows engineers to work on a "unified" vocabulary of requirement artefacts instead of using proprietary notations, since as shown in (Koch et al. 2006) these requirements artefacts can be translated into analysis models using model transformations.

In summary, MDE offers a suitable solution for Web requirements for several reasons:

1. It mainly focuses on concepts; the way to represent them is a secondary aspect. It offers a systematic way to translate requirements knowledge into the next phases in the development life cycle.
2. Additionally, as some relations are defined between requirements and analysis concepts, it can control the traceability and the coverage of requirements.
3. Finally, if an UML profile is defined for the requirements metamodel (as it is in WebRE), software support tools for modelling activities can be built in a cheap way.

2.2 Rich Internet Applications

The Web as it was originally conceived has dramatically changed since 2003 when the concept of Rich Internet Applications (RIA) appeared. This new kind of Web applications mixes the old navigation style of Web Applications with the behaviour of traditional desktop applications: client side feedback, drag and drop features, etc. Since then, almost any desktop application has a Web counterpart, allowing users to take advantage of automatic updates since no instalment is necessary at the client side. Some examples of Web applications with RIA behaviour are Google Maps (Google Maps), GMail and Google calendar (Gmail), Meebo (Meebo), etc.

As developers faced the same problems repeatedly and found good solutions using the concepts in RIA, some patterns arose. As in the design patterns field, different catalogues showing RIA solutions to abstract problems have been described; one of the most popular catalogues is the so called Yahoo Patterns catalogue (Yahoo Patterns). In contrast with software design patterns, RIA patterns are near to the stakeholder's perspective thus they use patterns' names when they describe specific RIA requirements. ADV-charts (Urbieta et.al. 2007) were proposed as a modelling approach to design the structural and behavioural of user interface (UI) elements of RIA applications. However their level of abstraction (close to implementation) is inadequate to be used during requirements specification.

In the following sections we describe how we extended WebRE to include RIA artefacts and how we derive tests from requirements.

3 A REQUIREMENTS META-MODEL FOR RIA

Expressing RIA behaviour and especially RIA patterns in requirements using a metamodel have many benefits such as:

- Making possible to develop the application easier by automatically deriving concrete software artefacts,
- Allowing the generation of tests to automatically validate the requirements
- Supporting requirements evolution and
- Improving traceability between requirements and the implementation.

In the following subsections we briefly present WebRE, its RIA extension and the corresponding UML profile.

3.1 WebRE

WebRE is a metamodel that includes modelling artefacts to deal with requirements in Web applications; it uses the power of metamodelling to fuse different approaches. WebRE was born from the exhaustive analysis of different Web engineering proposals that deal with requirements. It unifies the criteria of these proposals and defines a unified metamodel for the CIM (Computer Independent Model) level. It provides a base to decide which concepts or elements must be captured and defined in the requirements phase of Web applications. The metamodel defines each of these concepts and the relationships between them.

With this unification, WebRE overcomes an important gap: with the use of a common metamodel, it abstracts from the multiple notations used in each approach. Each artefact defined in WebRE can be mapped to an artefact in each different approach. Besides, WebRE also comprises an UML Profile with a concrete syntax to represent each concept. Thus, a development team can specify an application's requirements using the WebRE profile, and later (when necessary) map them to concrete model elements to continue with the selected methodology (NDT, UWE, W2000 or OOHDM). Additionally, it would be possible to systematically derive the corresponding navigation models from requirements expressed in WebRE using suitable transformations.

As a summary, WebRE provides a suitable environment for Web software development allowing the specification of navigation requirements of a Web application. However, WebRE was originally conceived for Web 1.0 applications and therefore it does not support specification of RIA behaviours. The extension proposed in this paper allows the systematic generation of models for Web 2.0 applications and the generation of tests to validate the RIA functionality (Section 4). In the following sub-section we show how we enriched the WebRE metamodel with new metaclasses and meta-associations in order to provide an approach that covers both: Web 1.0 and Web 2.0 requirements.

3.2 WebRE+

RIA has particular features like sophisticated interactive behaviour, client-side feedback of "slow" operations and different kinds of client-side behaviour depending on the occurrence of events, among others. An example of the last feature is shown in Figure 1. The line graph shows information about the progress of the business across the time. As a consequence of how progress is measure (it requires certain calculations) we only show the final computed value in the graph. The details of how those values were computed are shown only when the user shows interested in it (e.g. when the user puts its mouse over an item). This solution is well known as a hover detail pattern in the Yahoo Patterns catalogue. This kind of RIA behaviour improves applications usability without polluting the user interface with lots of information, which could be unnecessary at first sight. To provide a precisely specification of this kind of requirement we need to deal with concepts such as events, UI elements like buttons, textfields, etc. For this reason we extended the WebRE metamodel with these concepts as shown in figure 2.

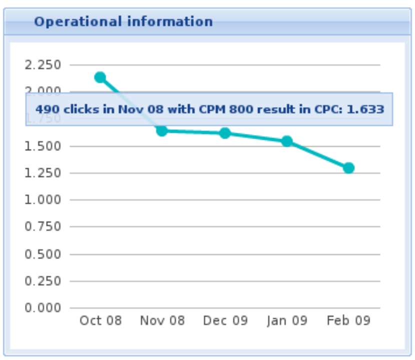

Figure 1: Hover detail pattern on a line graph

In WebRE+ the original packages, *structure* and *behaviour*, were kept to preserve the mapping between the concepts present in WebRE+ and its ancestors.

The structure package includes each concept to deal with the conceptual aspect of web requirements. Since RIA applications mainly deal with client side behaviour, we add the *UIElement* metaclass. Instances of this metaclass are: *buttons*, *textfields*, *images*, *checkboxes*, etc. To support RIA we extended the metamodel with two new metaclasses: *RIASpecification*, which represents a definition of a set of scenarios that a RIA behaviour must satisfy and *RIAScenarioSpecification*, which describes any *RIASpecification* in a concrete scenario. For example, in the hover detail feature (a *RIASpecification* instance), we must specify two different scenarios (*RIAScenarioSpecification* instances):

- Hover detail appears: When the user puts the mouse over an item then a *UIElement* must appear after 2 seconds. This *UIElement* must contain a name and a description of the item.
- Hover detail disappears: When the user moves the mouse out of the item then the *UIElement* with the details of the item must not be shown.

The *behaviour* package includes metaclasses to represent user's interaction and navigation. We extended the package with the *Event* metaclass which is important to specify different situations such as: when the user puts the mouse over an item, when the user types something on a field, etc. In this case, we differentiate between to different subclasses: those events which are originated with the keyboard (*KeyboardEvent*) and those which are originated with the mouse (*MouseEvent*). Also, we include a new metaclass *UIAction* which captures the actions that the user can perform over an element in the UI of the application (relationship between *UIAction* and *UIElement*). Instances of *UIAction* are "click", "type keys", and execution of one of the actions may produce many events, e.g. when typing a key on a user interface element three events are fired *onpressdown*, *onpresskey* and *onpressup*.

In the following subsection we describe our implementation of the UML profile in the Enterprise Architect environment.

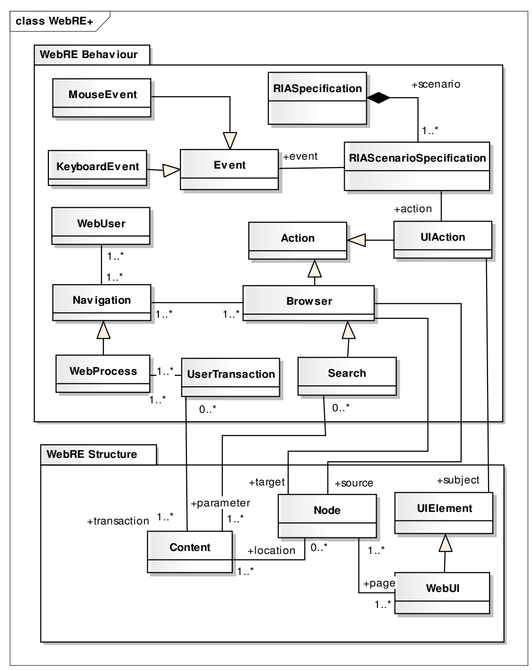

Figure 2: WebRE+ metamodel

3.3 A UML Profile for WebRE+

In order to provide editing support for our approach, we developed an UML profile for WebRE+, and implemented it using the Enterprise Architect tool. The use of UML profiles to provide tool supports is being used as a solution in some Web design approaches like UWE with MagicUWE (MagicUWE) or NDT with NDT-Profile (NDT-Profile).

In figure 3 we present the profile for WebRE+. As WebRE has its own profile, we only show our extension; that is, the metaclasses we have added to create WebRE+.

Each metaclass of WebRE extends an UML metaclass. Thus, we map our artefacts onto UML ones and define for them a set of characteristic that we could, even, improve with specific tag values or constraints.

In figure 4, we show an example work screen of WebRE+ in Enterprise Architect.

On the left, we can see a special toolbox for creating instances of the metaclasses. The user can select each WebRE+ artefact to deal with it in his diagrams. In figure 4, a WebUser instance (WebUserExample) and a RIASpecification instance (RIASpecification) example is presented. As RIASpecification is defined as an extension of the UseCase metaclass, it could be related with a User, like WebUserExample.

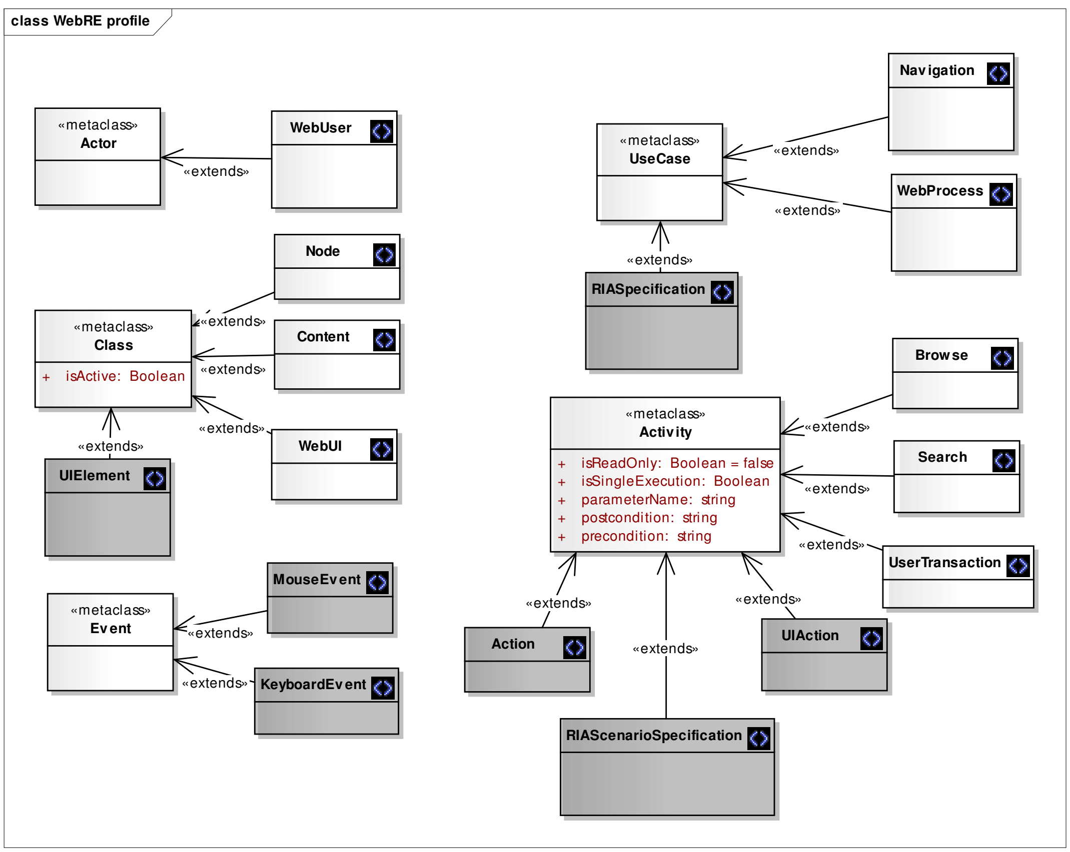

Figure 3: WebRE+ profile

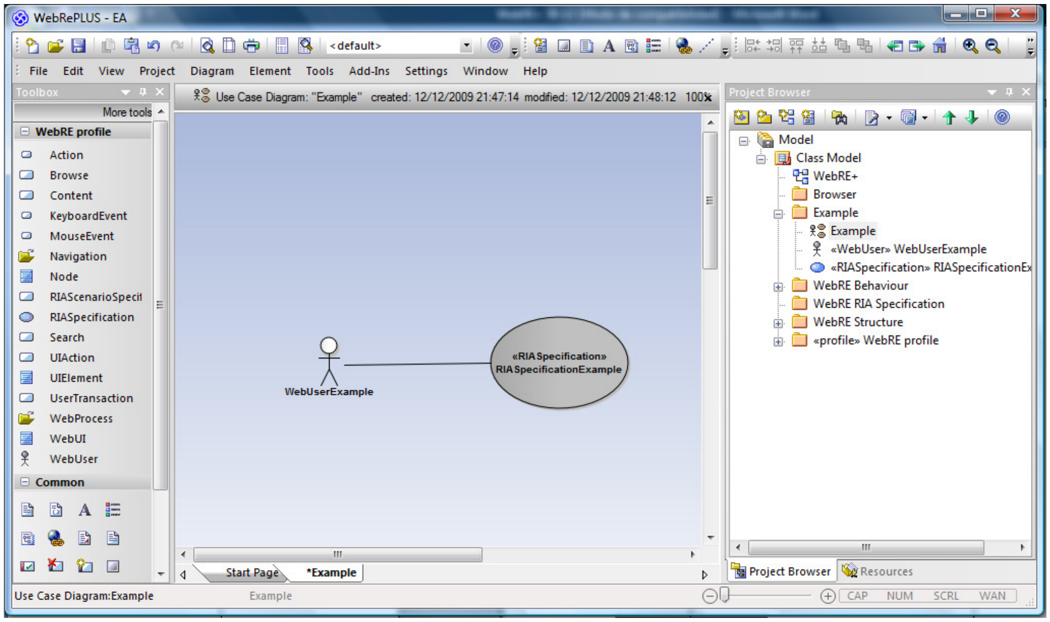

Figure 4: The WebRE+ profile in Enterprise Architect

4 TEST DERIVATION

Requirements validation is usually a hard and time consuming task which is performed by a quality assurance team after the application has been implemented. Generally, it is done manually (because requirements are captured using informal documents such as Use Cases or User Stories), by creating a set of tests that validate the requirements. The tests are run and if they pass, then the application can be deployed to production.

Using the formal definition that WebRE+ provides, we can use the requirement specification to derive these tests automatically thus reducing the time spent on the process and bridging the gap between requirements and tests. The process transforms a WebRE+-based model into a test model (Figure 5) that is independent of the platform. The transformation process follows these steps:

1. For each *RIASpecification*:
    a. Create a test suite.
    b. For each *RIAScenarioSpecification*:
        i. Create a test.
        ii. Add the actions of the scenario in the test.
        iii. Add an assertion for the post condition of the scenario.

The test model is then transformed into a concrete test implementation. So far, we have use Selenium (Selenium) for this purpose, although we could use a different framework such us Watir (Watir). We have chosen Selenium because it is one of the most popular testing frameworks that simulate user input and it is widely used in industrial settings. Also, a selenium test could be written in almost any programming language and run on a selenium server whereas Watir depends on Ruby.

In the following section we illustrate the use of the metamodel with a specific RIA requirement in the context of a Business Intelligence application showing how we specify it using WebRE+ and how tests are derived to Selenium.

5 A CASE OF STUDY

The business intelligence area is an example of how to use RIA to improve the user experience. For example, Pentaho BI suite (Pentaho) uses the Web environment to show data and allows users to execute queries to the data warehouse. A line graph that shows the progress of the business (Figure 1) could provide details about each value shown using the hover detail pattern.

Let's suppose that we are developing a Business Intelligence Web application for a company whose core business is organizing campaigns for different customers and providing summary reports to them. To improve the usability of the summary report which contains the line graph of Figure 1, we would like to add hover details to the items to show how those values are computed. For example, on a particular day there have been 3245 clicks and 15687 impressions so the CPC (Cost per click) is 0.34.

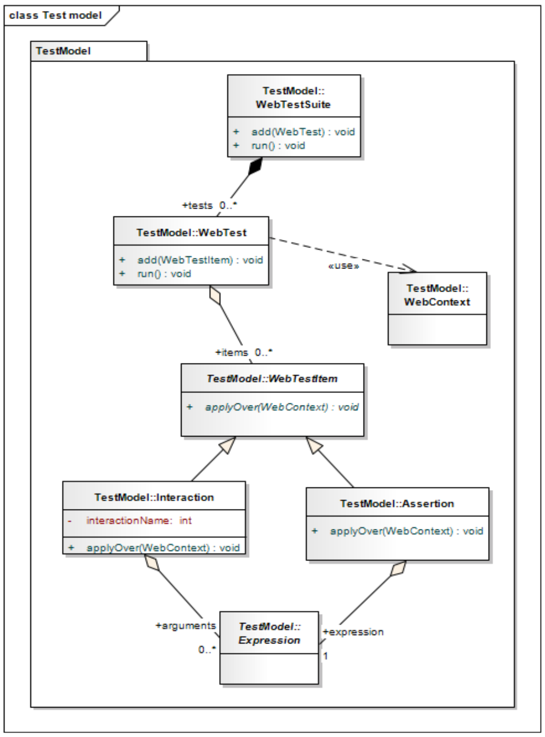

Figure 5: Test model

As on every RIA pattern, there are some features that can be configured and should be specified during the requirement elicitation phase. A simplified instance model of the WebRE+ specification for this requirement is shown in figure 6. The model shows that when the item receives an onmouseover event, a detail of the item must be shown in the page in less than 2 seconds. This widget must contain a label with the money used in the campaign and the number of clicks.

The WebRE+ instantiation describes the possible scenarios that the RIA behaviour must satisfy. Using the transformation explained in Section 4 we transform this model into an instantiation of the test metamodel and then we derive the test suite to the Selenium framework. The derived tests are shown next:

```
Test 1
(01) s.open(reportURL);
(02) s.mouseOver("id=item1");
(03) Thread.sleep(1000);
(04) assertTrue(s.isElePresent("id=d1"));
```

```
Test 2
(01) s.open(reportURL);
(02) s.mouseOver("id=item1");
(03) Thread.sleep(1000);
(04) s.mouseOut("id=item1");
(05) Thread.sleep(1000);
(06) assertFalse(s.isElePresent("id=d1"));
```

The test suite contains 2 tests, one for each scenario described in the WebRE+ model of figure 6. The first test opens the report (line 1), passes the mouse over the item (2) and waits till the item detail is shown (3), then the assertion verifies that the detail is present (4). The second test opens the report (1), passes the mouse over the item (2) and waits till the item detail is shown (3). Then the mouse is put out off the item and waits (4 and 5) and the assertion verifies that the detail is not present (6).

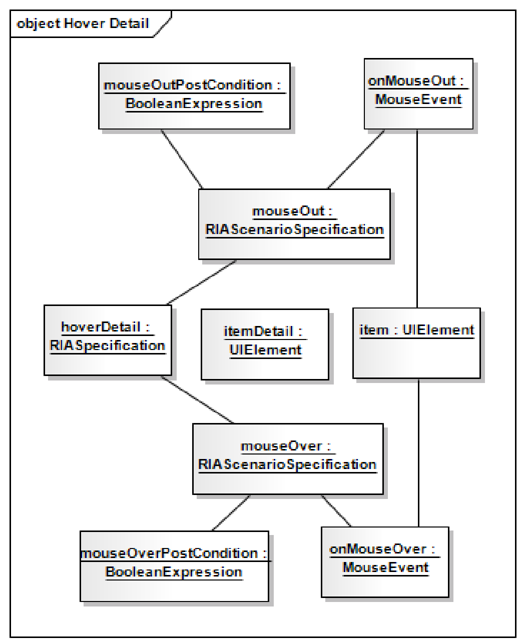

Figure 6. Hover detail's specification in WebRE+

6 CONCLUSIONS AND FUTURE WORKS

In this paper we presented a metamodel for capturing RIA requirements. The metamodel allows us to express different well known RIA patterns such as those in the Yahoo patterns catalogue. The metamodel has been implemented as a UML profile and used within the EA environment to capture different RIA requirements in the context of a business intelligence application.

We still need some further work on some of the motivations of using metamodels to capture RIA requirements. In this matter we are working on deriving part of the RIA functionality using well known Javascript libraries such us YUI or ExtJS. Finally, because of this kind of requirements not only affect the UI part but also the software backend, we are trying to indicate which part of the functionality could not be implemented automatically and thus needs manual intervention from developers.

ACKNOWLEDGEMENTS

This research has been supported by the project QSimTest (TIN2007-67843-C06 03)

REFERENCES

Atkinson, C., Kühne, T. 2003. Model-Driven Development: A Metamodeling Foundation. IEEE Software, 20(5), pp. 36-41.

Brambilla, M., Fraternali, P., Tisi, M. A Transformation Framework to Bridge Domain Specific Languages to MDEA. 4th Workshop on Model-Driven web Engineering. LNCS 5421. France, 2009. pp. 167-181.

Ceri, S. Fraternali, P. Bongio, A. 2000. Web Modelling Language (WebML): A Modelling Language for Designing web Sites. Conference WWW9/Computer Networks, 33(1-6), pp. 137-157.

Deshpande, Y., Marugesa, S., Ginige, A., Hanse, S., Schawabe, D., Gaedke, M., White, B. 2002. Web Engineering. Journal of web Engineering, 1(1), pp. 3-17.

Duhl, J. Rich Internet Applications. A white paper sponsored by Macromedia and Intel, IDC Report, 2003

Enterprise Architect. www.sparxsystems.com.au

Escalona, M.J., Torres, J., Mejías, M., Gutierrez, J.J., Villadiego, D. 2007. The treatment of navigation in web Engineering. Advances in Engineering Software. Vol. 38. pp.267-282.

Escalona, M.J., Aragon, G. 2008. NDT. A Model-Driven approach for web requirements. IEEE Transaction on Software Engineering, 34(3), pp. 370-390.

Escalona, M.J., Koch, N. 2007. Metamodelling the requirements of Web Systems. Web Information Systems and Technologies. Int. Conferences WEBIST 2005 and WEBIST 2006. Revised Selected Papers. LNBIP, Vol. 1, Springer, pp. 267-280.

Fons, J., Pelechano, V. Albert, M., Pastor, O. 2003. Development of Web Applications from Web Enhanced Conceptual Schemas. Conference on Conceptual Modeling LNCS, 2813, pp. 232-245. Springer Verlag.

Google Adwords. http://adwords.google.com. Last visit: 12/09.

Google Maps. http://maps.google.com. Last visit: 12/09

Gmail. http://www.gmail.com. Last visit: Dec 2009.

Jacobson I., Object-Oriented Software Engineering: A Use Case Driven Approach, ACM Press/Addison-Wesley, 1992.

Jeffries R, Andersen A., Hendrickson C., Extreme Programming Installed, Addison-Wesley, 2000

Koch, N. , Knapp, A., Zhang, G. 2008. UML-Based Web Engineering. In Web Engineering: Modelling and Implementing Web Applications, Springer. 157-191.

Koch, N., Zhang, G., Escalona, M.J. 2006. Model Transformations from Requirements to Web System Design. ACM International Conference Proceeding Series. Proceedings of the 6th international conference on Web Engineering (ICWE 2006). ISBN: 1-59593-352-2. Ed. ACM. pp. 281-288. Palo Alto. California. USA.

MagicUWE. http://uwe.pst.ifi.lmu.de/toolMagic
UWE.html.Last visit: 05/10.

Meebo. http://www.meebo.com. Last visit: Dec 2009.

Meliá, S., Gómez, J., Pérez, S., and Díaz, O. 2008. A Model-Driven Development for GWT-Based Rich Internet Applications with OOH4RIA. In Proceedings of the 2008 Eighth international Conference on Web Engineering (July 14 - 18, 2008). International Conference On Web Engineering. IEEE Computer Society, Washington, DC, 13-23.

Moreno, M., Romero, J.R., Vallecillo, A. 2007. An overview of Model-Driven web Engineering and the MDA. Web Engineering and web Applications Design Methods. Vol. 12. Human-Computer Interaction Series. Chapter 12. Springer.

NDT-Profile. http://www.iwt2.org/ndt. Last visit: 05/10.

Pentaho. http://www.pentaho.com. Last visit: 12/09.

Preciado, J. C., Linaje, M., Comai, S., and Sanchez-Figueroa, F. 2007. Designing Rich Internet Applications with Web Engineering Methodologies. In Proceedings of the 2007 9th IEEE international Workshop on Web Site Evolution (October 05 - 06, 2007). WSE. IEEE Computer Society, Washington, DC, 23-30.

Rossi, G., Schwabe, D. 2008. Modeling and implementing Web Applications with OOHDM. In Web Engineering: Modelling and Implementing Web Applications, Springer.

Selenium. http://seleniumhq.org Last visit: 12/09..

Urbieta, M., Rossi, G, Ginzburg J., Schwabe D.: Designing the Interface of Rich Internet Applications. LA-WEB 2007: 144-153

Vilain, P., Schwabe, D., Siechenius, C. 2002. A diagrammatic Tool for Representing User Interaction in UML. UML'2000. pp. 133-147. England.

Watir. http://watir.com/ Last visit: 12/09.

Yahoo Patterns, http://developer.yahoo.com/ypatterns/. Last visit: 12/09.

YieldManager. https://www.yieldmanager.com/. Last visit: Dec 2009.