The Triad-based Design of Rich User Interfaces for internet applications

Francisco Javier Martínez-Ruiz
Belgian Lab. of Computer-Human Interaction
Université catholique de Louvain
Place des Doyens, 1 – B-1348
Louvain-la-Neuve, Belgium
(+32) 10/478349
francisco.martinez@uclouvain.be – http://www.isys.ucl.ac.be/bchi

ABSTRACT
Current trends in web development still are attached to the web page paradigm. Nevertheless, new uses of already available technology and recent development in terms of concepts, as the asynchronous communication, have produced a new generation of web applications: Rich Internet Applications (RIAs). These web applications essays to fulfill user expectations in terms of usability, reliability, quality, maintainability and performance. In our work, we are going to present a designing methodology that pursued as goal describing and developing User Interfaces of RIAs in a standardized way.

Keywords
Rich internet applications, web engineering, Model driven engineering, USIXML

INTRODUCTION
The Development of Web User Interfaces remains an empirical exercise. Moreover, current designing tools are specialized in manual design. Nowadays, Model-based design of interactive software applications is an approach which is used in an increasing number of applications because of their capability to be build and the growing number of conditions that applications should fulfill e.g., Web applications are made for a wide spectrum of users from experts to novice ones. That's why their development must be guided by usability and ergonomic parameters in order to assure their quality. The Design of User Interface of web applications is not a trivial task. The most complex type of Web applications are the emerging Rich Internet Applications (RIAs). RIAs are Web applications that transfer most of the load of processing the user interface to the Web client while the predominant part of data (from control and maintaining to business data) remains on the application server [2]. A standard RIA architecture is shown in Figure 1.

EICS'10, June 21–23, 2010, Berlin, Germany.

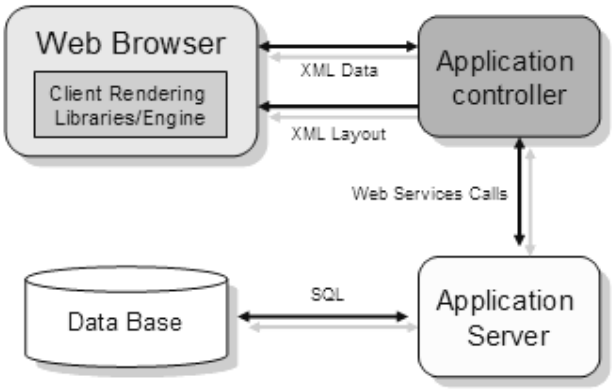

Figure 1: Typical architecture of a RIA application

A User Interface Description Language (UIDL) to denote a UI at any level of abstraction is not mandatory but is very useful. One of these languages is UsiXML (UsiXML which stands for User Interface eXtensible Markup Language). This language describes the UI for multiple contexts e.g., Character User Interfaces, Graphical and Multimodal ones in a form that maintains design independent from specific platforms [1].

THESIS STATEMENT
The research questions that guide our research are intimately related with the spirit of Model Driven Architecture and standardization, there is a need to develop a proper model for these emerging technologies to reduce developing costs and to produce flexible and adaptable interfaces for the next technological leap. Therefore, we inquire the following:

a. What are the elements that make different RIAs from Traditional Web applications?

b. Is it possible to extract from RIAs the common essential elements to model them in a neutral language?

c. What are the elements from RIA UIDLs needed to model today RUI frameworks features, such as: delivery of the interaction level, cinematic experience and multimedia elements?

Therefore, we will defend the following thesis:

"The introduction of a Conceptual modeling of Rich User Interfaces establishes a common ground to structure the development life cycle of the user interface of Rich Internet Applications according to a model-based approach and a step-wise approach which could be ported by a transformational schema to various web development environments".

MOTIVATIONS

Web applications are increasing their integration to our life. First, the document-based web was substituted by applications using the same channel but trapped in the same structure: the page metaphor. Now the classical web application is yielding her position to more complex and intuitive applications: Rich internet applications (RIA). This new kind of applications is emerging with the help of a plethora of toolkits and frameworks that help developers to create the sophisticated user interface that it's required by users since the beginning of web apps because of the natural comparison between desktop and web applications.

The Design of Rich User Interfaces for internet applications (RUIs) remains in the domain of experts who learned their craft over years. Some researchers [2] [9] are aware that there is a need of developing specific methodologies because the existing methodologies do not fulfill the challenges imposed by the RUIs among them we can list the followings. **(1)Lack of UI development methodologies**: as we stated before there are some methodologies but they were proposed to endorse the development of classical web applications, for instance: methods for creating a RUI are proprietary solutions or they are in initial states [2] [9]. **(2)Lack of experience in the UI development**: Every year, there are new emerging technologies, frameworks and tools for web applications. Each one requires a learning curve that is more pronounced by the lack of information about its integration, capabilities and usability considerations to the existing technologies. **(3)Lack of design knowledge for designing RIA UIs**: Developing a Web application requires gathering a vast knowledge in an important set of technologies which need to be carefully tuned to create in front of the user the illusion of unity. Most of developers couldn't achieve a full understanding of all the components and this is not even advisable because within development teams it's a better solution to specialize our personal. Lack of Model-based tool support: Proprietary and free tools are still scare for designing or coding of RUIs. Most of the tools are drawing or template oriented user development environments. **(4)Reaching known levels of consistency**: A user relies in his/her past experience with similar systems when using a new one (this is consistency, a well known principle of user interface design). Since the beginning of web apps the user's experience is behind the user expectations in comparison to desktop apps but now RUIs offers the possibility of reaching consistency of Web apps to known levels. Nevertheless, this implies a careful design, time and expertise of the developing tool. **(5)Usability Considerations**: developing a web user interface is a demanding task because we have to deal with browsers that don't follow W3C standards and usability is sometimes compromised to limitations imposed for creating a User Interface intended to a wide audience (usually this implies a design with simpler interfaces with fixed features).

RESEARCH METHODOLOGY

The research topics of this Thesis are UI Description Languages, Context Independence, the Web environment and its applications (webapps). The research area of this Thesis is relevant because of many reasons. First, it's a very expensive task for web developers the modification of webapps in order to be oriented to alternative targets. Current webapps do not support adaptability, usability and maintainability in a native way.

Research Methods

This Thesis uses as principal methodology, the Formulative research approach. That is, we are using as research method: The Conceptual analysis and The Concept implementation (proof-of-concept) [16] [17].

The research working process involves: (1) a review of literature in order to extract the state-of-the-art in a specific topic. (2) Incorporate new structures to the UsiXML language or propose alternative/new representations (RIA constructs). (3) Apply our proposal to different size or complexity case studies and then analyze the results. (4) Finally, a very concise and constrained prototype is going to be produced in order to have a proof-of-concept tool.

The validation of this Thesis is done in two complementary levels: First, a theoretical validation is completed by comparing our approach against others to respond the questions from the motivations and thesis statement sections. And second, empirical validation is conducted by a proof-of-concept implementation (a prototype design editor) which is applied to different case studies therefore validating its practical applicability.

GETTING STARTED

This section describes our method step by step modifying a toy case study in order to show all the updates. This chapter aims to give a panoramic view of the whole methodology with the purpose of establish a clear starting point for the reader.

Features of the Method

The development of web applications needs to be very flexible. Changes could appear in all the steps. In order to

deal with this characteristic, our method includes two important features: First, a **separation of concerns** structure and second is based in a **model driven approach**.

The general Method

Our method is a process of translation abstract models into more concrete ones (see Fig1) [OMG07]. The general process could be synthesized in the following steps: In the first phase, we create two models: a Task hierarchy model which describes User's goal and a Domain model to represent the data needed (T&D). The second phase implies the production of an Abstract User Interface (AUI) without any context or compromise with any technological platform. Then, in a Third phase a Concrete User Interface (CUI) is derived from the previous model, modality and platform widgets are decided. In the last phase, a Final User Interface (FUI) is obtained for a specific technology (For instance AJAX, .NET, LZX, SWF among others). This language-specific code is then rendered and executed in the web browser and we arrive to the operative RUI. In our work, we are going to present a designing methodology that pursued as goal describing and developing User Interfaces of RIAs in a standardized way. The name of this ensemble of models and meta-descriptions is, TRAIN (**T**riad-based **R**ich **I**nternet **A**pplication desig**N**).

TRAIN is a method for developing User Interfaces for RIAs. We start with an abstract definition of the UI. Then, in an iterative process more details are included until arriving to a concrete definition. Various features make TRAIN a viable choice: Extensibility (supports composition of structures), a set of Visual Patterns and the separation of concerns (data, logic and presentation). Next, we discuss the core concepts of our method which offer a scalable and model driven engineering approach [16] that is supported by the CAMELEON framework [5], UsiXML language [4], Zoomable User Interface Task hierarchies (ZUIT) and triplet-based design concept.

ZOOMABLE UI TASK HIERARCHY (ZUIT)

TRAIN defines the application as a hierarchy of tasks. This hierarchy is composed by multiple levels where inner nodes are gathering elements and leaf nodes are atomic tasks. Tasks are connected by temporal operators of three types (Sequential, concurrent and choice). Nevertheless, there is a shortcoming in these models (discussed in [13]). The complexity is directly proportional to the size of the application and at some point, icons and texts become unintelligible. Another problem is that models as ConcurTaskTrees notation (CTTs) (see Figure 5) which is discussed in [13], do not provide any semantic information through its structure since the structure is simply replicated at all levels. For instance, in a minimal Sign in application (see, Figure 1a) we have two tasks (Get information and Submit). First you have to recollect user data before submit it. Then, tasks are related by a sequential operator ([]>>). Get information is subdivided into two atomic tasks (Input User Name, Input Password) which are executed at the same time (|[]|).We use a treemap-like representation [14], instead of using an arborescence representation (Figure 1a) with an important variation: indeed, it is a Zoomable User Interface. The representation of the Sign in application is shown in Figure 1b.

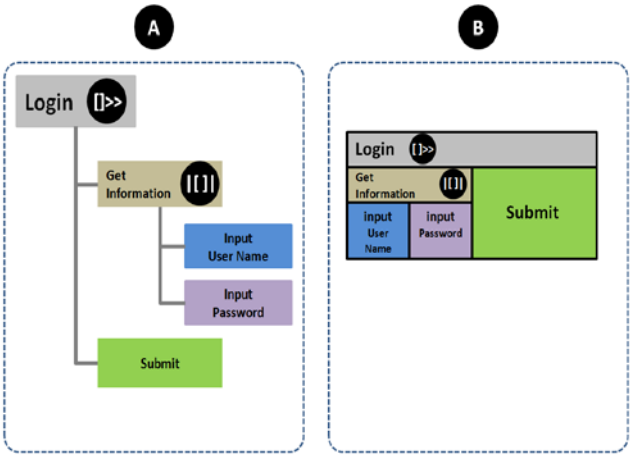

Figure 2: Login example in ZUIT format

TRIPET-BASED DESIGN

The idea behind our method is simple: Keep designers focused on their tasks so that they do not become distracted when they go though the development process. Their objective is to define the sequence of tasks that is needed to accomplish the application goal. For each task T introduced by the developer, task triplets (triads) are introduced in the model. T is substituted by T' under which R (Robustness) and D (Decorative) set of tasks are added. These groups exchange information between them and with the original task in order to update the model in terms of validation and presentation. In Figure 2b, this structure is represented as a ZUIT.

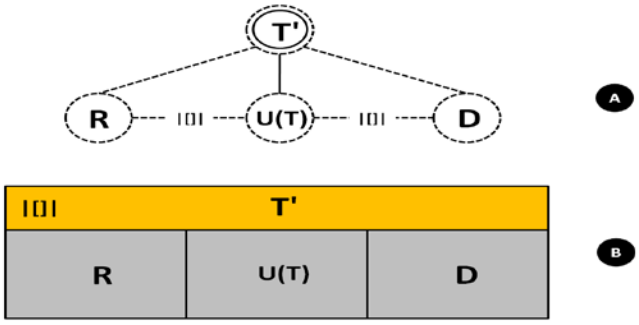

Figure 3 : Triplet of task model

Our model is loosely based in the triad of Vitruvius [8]. He stated that every building design should have three qualities: It must be strong, useful, and beautiful. We translate these features for UI design into: (1) Robustness tasks. This set agglutinates all tasks related to prevention, validation and recuperation. (2) Utilitarian tasks. This set includes Input, Output, Control and navigation tasks besides CRUD operations [9]. That is, all the possible operations to be executed over tasks. (3) Decorative tasks.

This set groups all tasks related to aesthetic aspects of the presentation of both, robustness and utilitarian tasks of UI.

CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

The goal of this work is to propose a MDA method for developing UIs of RIAs, which provides an ensemble of models in order to treat the complexity of RIA design. We have completed part of the proposed methods. For instance, the description of the Task Hierarchy as a Zoomable representation of the UI in order to avoid many of the disadvantages of typical task representation in UsiXML, such as tree node explosion and lack of structure [13]. Also, the design method helps developers to focus in their work instead of dealing prematurely with tasks related to operation and security. This is done by the introduction of the triad concept as integral part of development. More emphasis has been done in the task model in order to exploit it more in term of analysis of the container structure [9] and enrich it with a decorative RIA notation for AUI definitions which is proposed to introduce early in the development information about RIA-oriented tasks.

Future work

Many tasks are still a work in progress. For instance, our Repository of triad tasks is being updated with news tasks; Better behavior modeling; Adaptation to mobile technologies; as well as, a better understanding of the quality in the overall process; Better metrics in order to measure the task weights and exploit more semantic of ZUITs.

ACKNOWLEDGEMENTS

This work is supported by the Programme AlBan, the European Union Programme of High Level Scholarships for Latin America, scholarship No. (E06D101371MX) and the Belgian Computer Human Interaction Laboratory.

REFERENCES

1. UsiXML. http://www.usixml.org/ (May 10th, 2009)
2. Martínez-Ruiz, F.J., Muñoz Arteaga, J., Vanderdonckt, J., González-Calleros, J.M. (2006), A first draft of a Model-driven Method for Designing Graphical User Interfaces of Rich Internet Applications, Proc. of 4th Latin American Web Congress LA-Web'2006, IEEE Computer Society Press, 2006.
3. OMG, http://www.omg.org , (May 10th, 2009)
4. http://www.piccolo2d.org (May 10th, 2009).
5. http://jquery.com (May 10th, 2009).
6. Luyten, Kris, Clerckx, Tim, Coninx, Karin and Vanderdonckt, Jean M. (2003): Derivation of a Dialog Model from a Task Model by Activity Chain Extraction. In: Jorge, Joaquim A., Nunes, Nuno Jardim and Cunha, Joao Falcao e (eds.) Proceedings of the 8th International Workshop on Interactive Systems: Design, Specification, and Verification, Portugal.
7. Limbourg, Q., Multi-Path Development of User Interfaces, PhD thesis, University of Louvain, Nov. 2004.
8. Albin, S. T. (2003). The art of software architecture. John Wiley and Sons.
9. Martinez-Ruiz, Francisco J., Vanderdonckt, Jean and Muñoz, J. (2008): Web User Interface Generation for Multiple Platforms. IWWOST 2008. 7th International Workshop on Web-Oriented Software Technologies.
10. Bartalos, Peter & Bieliková, Mária: (S)CRUD pattern support for semantic web applications. Safarik University, Kosice, Slovakia (2008).
11. Montero, F., López-Jaquero, V., Lozano, M., González, P., IdealXML: un entorno para la gestión de experiencia relacionada con el desarrollo hipermedial, in ADACO: Ingeniería de la usabilidad en nuevos paradigmas aplicados a entornos web colaborativos y adaptativos, September 2005.
12. Carneiro, L. M., Cowan, D. D., and Lucena, C. J. 1994. ADVcharts: a visual formalism for interactive systems. SIGCHI Bull.26, 2 (Apr. 1994), 74-77.
13. Martínez-Ruiz, F.J., Muñoz Arteaga, J., Vanderdonckt, J., González-Calleros, J.M. (2009), Model Driven Engineering of Rich Internet Applications equipped with Zoomable User Interfaces, Proc. of 7th Latin American Web Congress LA-Web'2009 (Merida, Mexico), IEEE Computer Society Press, 2009.
14. Johnson, B. and Shneiderman, B. 1991. Tree-Maps: a space-filling approach to the visualization of hierarchical information structures. In Proceedings of the 2nd Conference on Visualization '91 (San Diego, California, October 22 - 25, 1991). G. M. Nielson and L. Rosenblum, Eds. IEEE Visualization. IEEE Computer Society Press.
15. Pederiva, I., Vanderdonckt, J., España, S., Panach, I., & Pastor, O. (2009). The Beautification Process in Model-Driven Engineering of User Interfaces. Dans Human-Computer Interaction – INTERACT 2007 (pp. 411-425).
16. Robert L. Glass, Iris Vessey, and Venkataraman Ramesh. Research in software engineering: an analysis of the literature. Information & Software Technology, 44(8):491–506, 2002.
17. JV. Ramesh, Robert L. Glass, and Iris Vessey. Research in computer science: an empirical study. Journal of Systems and Software, February 200