RIA: more than a Nice Face

Roberto Rodríguez-Echeverría

Quercus Software Engineering Group, University of Extremadura
Avda. de la Universidad s/n, 10071, Cáceres, Spain
rre@unex.es

Abstract. Nowadays, a new generation of Web applications built over Rich Internet Applications (RIAs) technologies is quickly attracting users eager for experiencing the richness of desktop applications together the incredible profits of networked available resources and services. RIA technologies can provide a complete rich user experience: richer and more interactive user interfaces supported by data and business logic at the client side and efficient bandwidth usage. Indeed, new and complex usage scenarios of those applications are devised, such as disconnected operation modes or multidevice (ubiquitous) access. Last RIA technologies bring new design concerns and issues increasing strongly the complexity of the development and evolution of that new breed of Web applications. However, only fledgling software engineering approaches have been proposed for RIA development, far from a complete and systematic methodology. Not only user interface concerns must be tackled, but adequate specification of data and business logic distribution and its consequences should be carefully considered. The main objective of this work consists on clearly identify and provide an unequivocal specification of the main issues and design concepts involved in the development of RIAs within data and business logic distribution concerns, eventually defining a complete and systematic methodology for model-driven RIA development.

Keywords: RIA, MDA, Disconnected Applications.

1 Problem Definition

Current Rich Internet Applications are highly complex systems characterized by the provision of a rich user interface as much as the distribution of data and business logic at the client side. Those core features provide a comprehensive rich user experience: rich user interaction, short respond time, efficient use of bandwidth and new capabilities as multimedia manipulation or disconnected operation modes.

As stated in [7], content and business logic distribution between server and client together a richer communication are main design concepts in the conceptual specification of RIAs. The authors of that work have specified them in general terms. However, due to their fundamental role in the development of Rich Internet Applications, a deeper and detailed analyze should be performed.

1.1 Data Distribution

In a RIA, as mentioned in [7], data can be classified based on its location and its duration in four types: volatile at the client side, persistent at the client side, volatile at the server side and persistent at the server side. To deal with such distribution, the communication between clients and server must be enriched. It is mandatory to keep data consistency, specifying adequate synchronization processes. According to the origin of that communication, basically, we can distinguish two types of synchronization processes: (1) server to client synchronization and (2) client to server synchronization.

On server to client synchronization, it should be specified which data must be replicated at the client side and when such replication must occur. Taking into account the client-server architecture of RIAs, normally, during the application loading a first set of data is replicated at the client side and, later on, new data is replicated at the client as a result of client or server requests. So, some approaches [8] propose solutions based on cache systems and asynchronous communication mechanisms (server push).

On client to server synchronization, the process increases in complexity strongly, especially when disconnected operation modes are allowed at the client side. In that scenario, a client can operate locally with its data delaying the synchronization process till the reconnection moment. Obviously, data conflicts may appear during those synchronization processes and they should be resolved before normal execution of the application can be resumed. On that kind of synchronizations, considering CRUD operations, data involved in creation, update or remove operations performed at the client side may be a source of data conflicts. On creation operations, as presented by [3], data conflicts may appear when it is required to keep a unique identifier of a piece of data on the whole system, e.g. in a distributed billing system in which disconnected clients are allowed to create their own bills. On update operations, data conflicts may appear frequently on scenarios of several clients trying to update the same piece of data.

In disconnected (offline) applications field [2][4], a wide range of solutions have been proposed to solve the aforementioned synchronization problems. Some of them present very restrictive scenarios defining data partitions and/or locks: in a given time only one client operates on a specific piece of data. While other ones present more flexible solutions based on the definition of complex politics to automatically reconcile data conflicts. Usage scenarios for all those approaches may appear in the development of RIAs, so they must be properly specified and modularized.

Another relevant factor on data synchronization is the data granularity to be used in transmission. That granularity has a clear impact in the efficiency of bandwidth usage, so the optimal data granularity must be selected in every transmission. In a RIA development method, there must be defined the mechanisms to specify the desired granularity of every concrete data transmission. As mentioned in [3], from an object oriented perspective, it should be possible to transmit an attribute of a data object, a

complete data object or a collection of data objects. Besides, data granularity depends on the type of operation performed on the data. On data deletion at the client side only the identifier field has to be transmitted. While, on update operations it will be necessary to send the identifier and all the name/value pairs of the modified attributes.

1.2 Business Logic Distribution

Regarding business logic localization, as stated in [7], we can basically consider three different cases: (1) business logic performed at the client side, (2) business logic performed at the server side, and (3) mixed (operation chains mixing operations at both sides). Another factor to take into account on the specification of business logic distribution is the multidevice perspective of RIAs. The devices and contexts at the client side may be really heterogeneous: from a PC in a wide bandwidth network to a mobile with an intermittent connection in a restricted bandwidth network. The specific characteristics of the client context (device, technology and network) should be carefully considered during the specification of the business logic subsets to be distributed in a concrete client.

Again, granularity appears as a main factor to be carefully considered. The granularity of business logic distribution has a clear impact on the adaptation flexibility. In that sense, it should be possible to distribute coarse-grained tasks or fine-grained operations and actions composing a larger task.

Additionally, the application may allow the client to switch to an offline operation mode. In such a case, the business logic distribution at the client side may be affected by this operation mode switching. Usually, a larger task subset will be specified to be distributed at the client side as a response to an event of switching to an offline operation mode from that client. And that situation may be wanted to be reverted back in the client reconnection.

In conclusion, the specification of business logic distribution in a RIA must permit to define different task subsets to be replicated at the client side, according the target device features, the connection network characteristics, the RIA technologies used and the operation modes considered (online and/or offline).

2 Motivation

The motivation for developing this research is three-folded: (1) RIA development is a hot topic in professional and academic worlds, (2) there is a lack of a complete and systematic methodology for RIA development, and (3) it is aligned with the main interests of the software engineering researching group I am member of.

Nowadays, everyday our lives depend a little bit more on that kind of applications and technologies based on the "network-as-a-platform" vision. Every now and then new RIAs appear featuring richer ways of interaction among its users and the use of richer

information sources (mash-ups). Those applications are changing our ways of working (Google docs, RTM), our means of learning and our communication channels (YouTube).

While RIA technologies are gaining maturity enough and there are a wide and increasing number of Web applications based on them, complete models and methodologies for the systematic development of RIAs are required. Currently, most of the aforementioned issues are treated and solved ad-hoc by developers, adding extra complexity to the development and management of RIAs. Moreover, reusing those solutions on the development of other RIAs is practically discarded due to its unsystematic nature. In the last years, interesting approaches had been proposed mainly focused on the specification of RIA presentation concerns [9][5][6][7]. However, none of them allow expressing data communication granularity, conflict reconciliation policy, multidevice business logic distribution, among other RIA development related issues.

And, finally, this work is defined as an extension of the work performed inside the RUX Method Project. This extension is conceived as a transmission of the RUX Method principles to the Data and Business Layer specification within a complete, systematic and model-driven methodology for the development of Rich Internet Applications.

3 Objectives

The main objective of this work can be summarized as defining the modeling concepts, methods and tools to capture the specific features of the development of RIAs, focusing on the specification of the data and business logic distribution and communication concerns. This work should be seen as a single piece inside a major goal as it is the definition of a complete, systematic and model-driven methodology for the development of Rich Internet Applications.

The model-driven essence of the approach to be performed can be seen as a subsidiary goal. Indeed it is intended this approach should be localized inside the OMG MDA framework [1], providing models, methods and tools based on the MDA-related technologies[1].

4 Research Methodology and Roadmap

Regarding the research methodology, we think it should be adequate to use an iterative and incremental researching process conformed by four fundamental phases: (1) knowledge acquisition, (2) conceptual approach proposition, (3) conceptual approach application and (4) development of actual systematic methods and tools.

[1] http://www.omg.org/mda

During the first phase, two fundamental missions should be accomplished: (1) identify specific issues regarding data and business logic distribution in RIA development and (2) evaluate how accurately current RIA approaches cover them. For the first one, a thoroughly study of RIA technologies and its design patterns must be performed. In the meantime, the most active RIA development approaches from Model-Driven Web Engineering field must be constantly reviewed and analyzed.

Based on the knowledge acquired in the first phase, a conceptual approach should be propose or refined in the second phase. Using the technologies of the OMG MDA framework, that approach must define the elements and methods to permit the correct specification of the concerns elicited in the first phase.

Once completely defined a tentative conceptual approach, a comprehensive study case (the application example) should be manually developed driven for our conceptual approach in order to validate its correctness and to measure its accuracy on the specification of the concerns elicited in the first phase.

The main goal of the last phase consist on applying all the knowledge and experience acquired along the previous phases in the development of systematic methods and tools to ease the utilization of the conceptual approach in the development of full fledged RIAs.

Regarding how iteration happens, when a problem or a missing is detected during the evaluation of the conceptual approach carried out during phase 3, the process can switch to phases 1 or 2 depending on the nature of the problem detected.

Currently, we are deeply involved on the identification of the issues related to RIA development, focusing on the data and business logic distribution concerns, while constantly reviewing other RIA development approaches.

5 Main Contribution

The main contribution of this work to Web Engineering can be summarized as clearly identifying and providing an unequivocal specification of the main issues and design concepts involved in the development of RIAs within data and business logic distribution concerns. As mentioned before, this work is just a stepping stone on the more ambitious goal of providing a complete and systematic methodology for model-driven development of full fledged Rich Internet Applications.

Acknowledgment. This work is partially supported by MICINN under contract TIN2008-02985 and by Junta de Extremadura under contract GRU09137.

References

1. Jean Bézivin and Olivier Gerbé. Towards a precise definition of the omg/mda framework. In ASE '01: Proceedings of the 16th IEEE international conference on Automated software engineering, page 273, Washington, DC, USA, 2001. IEEE Computer Society.
2. Maria A. Butrico, Henry Chang, Anthony Cocchi, Norman H. Cohen, Dennis G. Shea, and Stephen E. Smith. Gold rush: Mobile transaction middleware with java-object replication. In COOTS, pages 91-102, 1997.
3. Pawel Gruszczynski, Stanislaw Osinski, and Andrzej Swedrzynski. Offline business objects: Enabling data persistence for distributed desktop applications. In OTM Conferences (2), pages 960-977, 2005.
4. Avraham Leff and James Rayfield. Programming model alternatives for disconnected business applications. IEEE Internet Computing, 10(3):50-57, 2006.
5. Sandy Pérez, Oscar Díaz, Santiago Meliá, and Jaime Gómez. Facing interaction-rich rias: The orchestration model. In ICWE, pages 24-37. IEEE, 2008.
6. Juan Carlos Preciado, Marino Linaje Trigueros, Rober Morales-Chaparro, Fernando Sánchez-Figueroa, Gefei Zhang, Christian Kroiss, and Nora Koch. Designing rich internet applications combining uwe and rux-method. In ICWE, pages 148-154. IEEE, 2008.
7. Juan Carlos Preciado, Marino Linaje Trigueros, and Fernando Sánchez-Figueroa. An approach to support the web user interfaces evolution. In AEWSE, 2007.
8. Jie Song, Ge Yu, Daling Wang, and Tiezheng Nie. Offline web client: Approach, design and implementation based on web system. In WISE, pages 308-314, 2006.
9. Matias Urbieta, Gustavo Rossi, Jeronimo Ginzburg, and Daniel Schwabe. Designing the interface of rich internet applications. In 2007 Latin American Web Conference (LA-WEB 2007), pages 144-153, Washington, DC, USA, 2007. IEEE Computer Society.