

THE EVOLUTION OF USER INTERFACES IN THE CONTEXT OF RICH INTERNET APPLICATIONS

Napoleon Alexandru SIRETEANU

Faculty of Economics and Business Administration
Alexandru Ioan Cuza University
Iasi, Romania
salexis_ro@yahoo.ro

Abstract

Desktop Applications have evolved to websites, web applications and rich internet applications. Web technologies have been changed in the last years, because they adopted the richness of the desktop applications. Rich Internet Applications (RIAs) support many features of a classical desktop application, offering a "rich user experience" based on a state user interaction. Rich Internet Applications provide advanced features such as direct object manipulation and animations that can enrich the user's experience. This article discusses some aspects of designing user interfaces for modern software applications and the technologies used in the RIA context.

Keywords: software applications, web applications, Rich Internet Applications, usability

JEL classification: L63, L86

1. INTRODUCTION

The times of classical (offline) desktop applications and simple informational websites are coming to an end. Modern applications combine the richness of the desktop with the network of Internet services.

In concrete terms, Web 2.0 manifests itself as video-sharing sites, wikis, blogs, and communities - technologies that enable user-generated content across a variety of media and forms - and RSS feeds, podcasts, and applications such as Google Maps - technologies that improve user access to and interaction with the wide variety of media being generated [Blank, 2008, 4].

One of the biggest advantages of Web 2.0 is the rise of new possibilities to improve the usability of web software. Designing the interface of modern Internet applications (rich internet applications - RIAs) is difficult to achieve because they must combine hypermedia interfaces from traditional web software with the interface functionality we find in desktop applications, like drag and drop and pop-up windows.

2. EVOLUTION OF DESKTOP AND WEB USER INTERFACES

Desktop Applications are usually mentioned to us as applications that need to be installed on an operating system. A classical Desktop Application is, for example, a spreadsheet application such as Microsoft Excel, or a graphical editor like Adobe Photoshop. *Internet-enabled applications* are applications that use network support, but also run offline (for example an e-mail client such as Mozilla Thunderbird or Apple Mail). Very similar to those are *Rich Clients*, where an important part of the application's features work on the user's local system. On the other hand, there are classical *websites*, *web applications* and *thin clients*, which can be started and loaded via the network.

The technologies described by the term *Rich Internet Applications (RIA)* are situated between Web-related and Desktop-related technologies. Some run inside a web browser, others without, but a common characteristic aspect is that RIAs benefit from the best of usual Desktop Applications and of Web applications.

Web browser-based RIAs are websites with more richness of user experience, more response and personalization facilities, calendars like "Google Calendar" or online email services like "Yahoo! Mail" used via a browser (both applications are developed in Ajax).

There are also some applications that look and feel like a desktop application, because of their rich and responsive user interfaces. One example of a *desktop-related RIAs* is Apple iTunes, that needs to be installed locally but also acts like a shopping website and an application for communication with handheld music/video devices like the "Apple iPod" or the "Apple iPhone" smartphone.

2.1 Evolution of User Interfaces

Due to the proliferation of Web applications that offer a rich user experience, traditional Web applications have been replaced by the Rich Internet Applications (RIAs), which offer an interface, interaction and functionality capabilities similar to desktop applications.

According to Jakob Nielsen, the generations of user interfaces are [Nielsen, 1993]:

- generation 0: without graphical user interfaces, direct hands-on access to the hardware; zero dimensional systems;
- generation 1: pioneer system user interfaces, which are non-interactive; zero dimensional systems;
- generation 2: line-oriented/batch interfaces used on mainframe computers, even with simultaneous user access; one dimensional user interface;
- generation 3: traditional full-screen with a possibility of modifying the whole screen; two dimensional interfaces. The user interfaces are mostly character-based, surrounded with poor graphical objects;
- generation 4: modern graphical user interfaces (known as WIMP - windows, icons, menus and pointing devices); two-and-a-half dimension interfaces. The interaction is mainly based on direct manipulation with keyboard and mouse; this is the beginning of the desktop applications;
- generation 5; future (beginning with 1995) or the next generation interfaces; Nielson suggested that the dimensions will rise with multimedia elements like sound, voice and virtual reality.

In a progressive approach, from 1995, GUIs are mainly based on the "desktop representation", which lies on the idea of the Xerox PARC company that developed a commercial

product called “Xerox 8010” or in short “Xerox Star”, in 1981. This was the inspiration for the first commercial successful desktop based operating system by Apple, the “Apple Lisa”. The most prominent desktop-based UIs from the beginning until today are “Microsoft Windows”, “Apple Mac OS” or “Linux X Window” based implementations such as “KDE” or “GNOME”. The concept of a WIMP desktop hasn’t really changed until today.

RIAs eliminate multi-screen interface, offer single application view and reduce iterations in business processes. RIAs use the client’s processor, offer real-time user-interface options, things that are not possible with the standard HTML used in browser-based Web applications.

RIAs provide a user interface that is more robust, non-dependent on platform, responsive and visually attractive than the one that is achieved with standard technologies (HTML, CSS), and allow users to interactively visualize and manipulate complex data more effectively and easily.

2.2 Usability

When we talk about software applications and their corresponding user interfaces, usability plays an important role for general orientation, the manner in which user interfaces are designed, organized and built. According to ISO 9241-11, the definition of usability is “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” [ISO, 1998]. This document states that a software product with a graphical user interface needs to provide ways to get a richer user experience.

In designing the look of an application instead of relying on the richness in user experience, the web has some advantages. With web technologies, it is easier to achieve design goals, because the technology with HTML and CSS is truly made to design the look of a graphical user interface.

3. THE EVOLUTION OF WEB TECHNOLOGIES

Web technologies have become richer over the last years and, gradually, they adopted the richness of the desktop applications.

The early World Wide Web was introduced by Tim Berners-Lee in 1991 in the form of an application that published simple HTML pages on a web server. A multi-platform browser (see Figure 1) is used to display pages [CERN, 2008]. The first Web application is an interactive website (a phone book) [Shklar, 2003, 5].

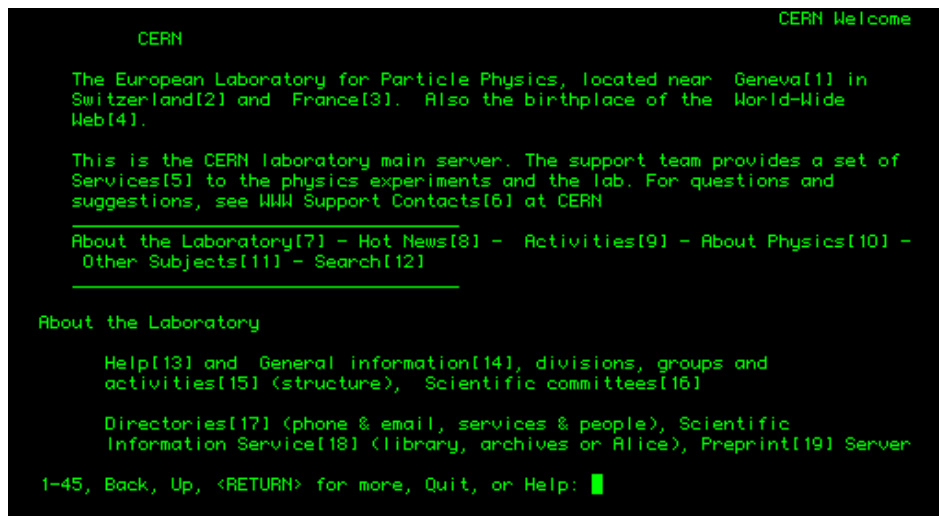


Figure no. 1 The first universal line-mode browser

The WWW was based on the following three standard concepts:

- URI (Uniform Resource Identifier)
- HTTP (Hypertext Transfer Protocol)
- HTML (Hypertext Markup Language).

In 1994, the first version of the Netscape browser has been announced, written by the inventors of the Mosaic browser. JavaScript, a client-side programming language, with a syntax similar to Java, brings new and richer functionality to web pages and improves the interface of web applications. The modification and access to the structure of a document is based on DOM (Document Object Model). In 1997, DHTML (Dynamic HTML) offered more HTML tags and parts of the W3C (World Wide Web Consortium) standard CSS 1 (Cascading Style Sheets). In combination with the CSS 2 and JavaScript 1.2, document components could be changed in style, position and visibility.

In 2001, Internet Explorer 6 was released, but CSS support was worse compared to the Gecko Engine. The Gecko Engine is a layout engine for HTML and CSS rendering. Based on the Gecko Engine, an Open Source project of Netscape, the Mozilla project appeared. The Mozilla Group released the very successful browser Mozilla Firefox that has become the most used browser. With Mozilla, programmers have the possibility to really join in the browser development for the first time. Google Chrome comes from behind very quickly and is the main competitor for Firefox. Safari was first released in 2003 by Apple Inc. and is based on the default Linux browser Konqueror.

High bandwidth, consumer demands and new technologies (including Web 2.0) have contributed significantly to the proliferation of RIA since 2004. RIA offers significant benefits for users and the most popular examples of frameworks include: Adobe Flex and AJAX, Google Earth, Google Mail and Google Finance. In present days, technologies such XHTML, CGI forms and scripts are replaced by approaches such as AJAX and Web services.

3.1 Rich Internet Applications (RIA) and rich user experience

The term Rich Internet Applications (RIA) was used for the first time in the Macromedia white paper “Macromedia Flash MX - A next-generation rich client” by Jeremy Allaire in 2002. “Macromedia Flash”, today “Adobe Flash” was called RIA. Interactivity, responsiveness and richness are the most important characteristics of RIAs. They offer support for validation and error handling, drag-and-drop functionality and richer controls like calendars and sortable lists. With the support of AJAX, web applications do not have to reload the whole web page, but can give instead a real-time search, sorting and filtering feature.

RIA should support many features of a classical desktop application, offering a “rich user experience” based on a state based user interaction, which means that an application resides in different states and can suggest this status and progress of the task, and respond to user commands. Rich user experience aspects distinguish themselves through a higher interactivity and responsiveness. The rich user experience is the key feature and the main motivation for using RIAs.

Of the RIA platforms (Flash, Silverlight and JavaFX), Flash has the broadest market coverage and is the winner regarding elaborate graphical user interfaces, as well as for resolution independence concepts. Silverlight may be established, especially among Microsoft environments, because Microsoft can push the distribution on Windows and Internet Explorer, which are both leaders, one on an operating system and the other on the Web browser market.

3.2 Semantic Web and Semantic RIA

The *Semantic Web* is, from today's perspective, a vision of a future Web, a vision first expressed by Tim Berners-Lee, the inventor of the WWW. The Semantic Web is based on technologies that enable better artificial intelligence to analyze and share data. The semantic web is an extension of the current web, where information is better accessible, better understood by machines and where the cooperation of people, who produce content, is much higher.

The idea of a Semantic Web is using XML for data storage and RDF (Resource Description Framework) to add meaning to data. This meaning is coded in sets of triples, consisting of a subject, a predicate and an object. The metadata model of the Semantic Web can be transferred to the Desktop environment, called *Semantic Desktop*. The Semantic Desktop is just an extension of the Semantic Web idea. The resources of the Semantic Desktop are not only web content, but all information (documents, multimedia and messages) stored in a single device, which is mostly a personal computer. According to Sauermann, Bernardi and Dengel, “The Semantic Desktop is an enlarged supplement to the user's memory” [Wikipedia, 2011].

In present, the current search engines (Google, Yahoo!, Bing) are not capable of indexing the plugin-oriented RIAs, such as Flex, OpenLazlo or Silverlight and, in consequence, prevents users from finding them and reading their contents. To overcome these deficiencies and provide users a better experience, a new type of RIA is required, called *Semantic RIA*, which extensively uses Semantic Web technologies for overcoming the problematic issues mentioned above [Hermida, 2011].

3.3 Client-server paradigm in the RIA context

Rich Internet Applications have one fundamental characteristic: one part is processed on a client machine and another part on the server-side, but which and how much processing will be distributed to which side depends on several aspects.

Table 1 shows a separation of application layers that are either processed on the client side or on the server side. If many layers of a client-server structure are placed on the client, it is called *fat* or *rich client*. If only some layers are placed on the client, we talk about *thin clients*. We can separate RIA applications in four columns - (2) to (5) like in Table 1 - from fat on the left to thin on the right. All this methodologies from (2) to (5) will be in the RIA context; it doesn't matter if they work on web or desktop. The elements from the middle are called "hybrid". For a global overview, two extreme positions are added in Table 1: on the left, the conventional desktop application (1), and on the right a traditional static web page (6).

Table no. 1 Client Server Architectures in the RIA context, adaptation from [Domenig, 2011]

	Desktop application	Fat/Rich client	Hybrid	Hybrid	Thin client	Static webpage
Client	RIA (Rich Internet Application)					Browser
	Presentation Logic Business Logic Logic Data	Presentation Logic Business Logic Logic	Presentation Logic Business Logic Logic	Presentation Logic	Presentation Logic	
Server			Business Logic Data	Presentation Logic Business Logic Data	Presentation Logic Business Logic Data	Presentation Logic Business Logic Data
	1	2	3	4	5	6

The separated elements represent mainly three parts:

- data
- business logic
- presentation logic/visual layer.

In usual desktop applications, there is nothing to be processed on a remote server. With a fat or rich client, only the data, mostly in a database, is located on a remote server. This fat-related hybrid solution in contrast to (2) differs in the distribution of business logic between client and server. Special parts of processing can be executed on the server or on the client side. This next thin related hybrid solution differs from (3) in the distribution of presentation logic between client and server. With a thin client approach, data and business logic are located on the server, and also a part of the presentation logic. The presentation engine is the only thing located on the client. This engine just displays the GUI, the logic is stored and processed remotely. The traditional static webpage solution offers a good approach for non-interactive systems.

4. DEVELOPERS PERSPECTIVE OVERVIEW

On the web, scripting languages are interpreted in the following sense: their source code is executed straight from text files, immediately after parsing. There are server-side scripting languages, such as Perl, PHP or Python, and client-side scripting languages, such as JavaScript executed by the web browser. The advantages of scripting languages are that the development costs are lower; they offer faster results, and more rapid and easier development. Scripting language-based web applications are typically achieved with procedural programming instead of using an object-oriented paradigm, which would offer better abstraction but requires more work in designing the system.

Desktop applications are usually built using C, C++, Java, or the Microsoft alternatives (including C# .NET). They usually implement the two-tier client-server communication model. They often make use of native calls to the operating system and its hardware drivers [Tiwari, 2010, 406]. The usage of combined libraries requires less time and work for the implementation of software.

Another principle for large scale software is modularity, which offers a separation of problems and a separation of responsibilities for team members. Automated versioning used with general-purpose languages is just another feature for large projects. These approaches are more complex, but enable developers to accomplish and manage the complexity of large systems.

In principle, for small projects, scripting languages are the cheapest choice, because immediate results can be achieved with them, but the use of general-purpose languages makes sense for larger projects.

The union between web and desktop could be seen with the PHP scripting language: initially developed as a processing language for web forms, but from 1998, applications developed in PHP got larger and they need an object-oriented syntax and a more consistent language. Another popular example that can be mentioned is Macromedia Flash, known as Adobe Flash at present. Flash was initially based on a graphical IDE, but, with the introduction of ActionScript, it became a fully recognized programming language in 2000. Object-oriented programming (OOP) was added by ActionScript 2.0 in 2004 and in 2006, ActionScript 3.0 introduced a better compiler and better error handling for developers. Flex has also been introduced and the scripting language paradigm became richer and richer.

The RIA space is red hot with technologies and competition, but Adobe continues to prove it is just a few steps ahead. HTML web applications will always have a place, but the industry is moving forward [Ahmed, 2011, 14].

5. VISUAL DISTINCTION BETWEEN DESKTOP AND WEB APPLICATIONS

Web is the universe for more personalization and the desktop provides more standardization. Web sites and web applications have a strong graphic styling or branding and the users are very familiar with different styles. Desktop applications with no explicit style have the same appearance like the native GUI of the operation systems or like the “Look and Feel” given by the platform they are implemented for. On the desktop, a user has the possibility to give the whole operating system another style. On the Web, a universal styling shared by several websites does not exist.

Desktop applications are still more standardized in their visual appearance than websites, but desktop applications or Desktop-related RIAs increasingly adopt the Web styling.

The styling becomes a very important aspect, as Figure 2 of “The New York Times” Flex application illustrates.

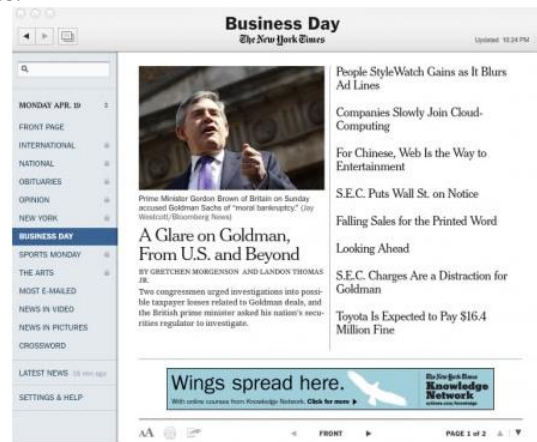


Figure no. 2 New York Times - Adobe Flex application

As a Web-related RIA, Adobe Flash offers a standard set of user interface components and a ‘Look and Feel’ concept for a unified skinning approach. Skinning is the change of the look of software. With the rise of RIAs and the convergence of Web and Desktop, personal skinning becomes more and more important for RIAs.

6. CONCLUSION

RIA promises to offer the best of both Web and Desktop, because web application development trends show that each one adopts the best features of the opponent. So, finally, both join in a way that makes them combine into one consistent concept. The rich user experience is a key feature for RIAs; AJAX appeared because of this and it offered a revolutionary possibility inside the browser environment. A move to general-purpose programming languages can be recognized, especially for back-end development. For front-end development and design, new scripting languages appear, which qualify for best interaction with back-end languages.

User interface design is a variable that is changed constantly due to the new concepts and technologies. User-centered design activities remain essential in order to evaluate the strengths and the weaknesses of new technologies.

Acknowledgement

This work was supported by the project "Post-Doctoral Studies in Economics: training program for elite researchers - SPODE" co-funded from the European Social Fund through the Development of Human Resources Operational Programme 2007-2013, contract no. POSDRU/89/1.5/S/61755.)

References

- [1] Ahmed, T., Orlando, D., *Flex 4 in Action*, Manning, 2011

- [2] Blank, R., Otuome, H., Gonzales, O., Charlton, C., *AdvancED Flex Application: Development Building Rich Media X*, Apress, 2008
- [3] Domenig, M., *Rich Internet Applications and AJAX - Selecting the best product*, at <http://www.javalobby.org/articles/ajax-ria-overview/>, accessed on Nov 12, 2011
- [4] Hermida, J., Meliá, S., Montoyo, A., Gómez, J., *Developing Semantic Rich Internet Applications Using a Model-Driven Approach*, published in *Proceeding WISS'10 Proceedings of the 2010 international conference on Web information systems engineering*, Springer-Verlag Berlin, 2011
- [5] Nielsen, J., *Noncommand User Interfaces*, at <http://www.useit.com/papers/noncommand.html>, 1993, accessed on November 9, 2011
- [6] Shklar, L., Rosen, R., *Web Application Architecture. Principles, protocols, and3 practices*, Wiley, 2003
- [7] Tiwari, S., Elrom, E., *AdvancED Flex 4*, Apress, 2010
- [8] *** *Semantic desktop*, at http://en.wikipedia.org/wiki/Semantic_desktop, accessed on May 17, 2011
- [9] *** *The first universal line-mode browser*, at <http://info.cern.ch/LMBrowser.html>, 2008, accessed on February 15, 2011
- [10] *** *User-centered System Design*, ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability, at <http://www.it.uu.se/edu/course/homepage/acsd/vt09/ISO9241part11.pdf>, First edition, 1998, accessed on January 9, 2011