

STUDY AND IMPLEMENTATION OF VARIOUS RICH INTERNET APPLICATION TECHNOLOGIES AND ITS BECKONING POSSIBILITIES

Sabbir Hossain
Student ID: 04101006

Department of Computer Science and Engineering
January 2007



BRAC University, Dhaka, Bangladesh

DECLARATION

I hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Signature of
Supervisor

Signature of
Author

ABSTRACT

For the past ten years- the technological advancement on the field of Internet Applications or Online Services has been astonishing. From the last few years of the previous century to this day, the World Wide Web has gone through an astounding overhauling. Web services have expanded to greater heights, technology has dug into deeper depths. This paper will give the reader an idea about how this new era of applications- Rich Internet Applications- surfaced on the World Wide Web and will also delve into the various technologies that are frequently being used in this field. It will also go into details about a project that is created by using these technologies. The end of this paper will then analyze the effect of Rich internet Applications- how the perception of local and online application is changing and the possibilities that are glimmering on the horizon.

TABLE OF CONTENTS

	Page
TITLE.....	1
DECLARATION.....	2
ABSTRACT.....	3
TABLE OF CONTENTS.....	4
LIST OF FIGURES.....	6
CHAPTER I. INTRODUCTION	
1.1 Overview of this Paper.....	7
1.2 The Title Explained.....	7
CHAPTER II. RICH INTERNET APPLICATION	
2.1 The Internet.....	8
2.2 Comparative Analysis: Web 1.0 VS Web 2.0.....	8
2.2.1 Web 1.0 defined.....	8
2.2.2 Introducing Web 2.0.....	11
CHAPTER III. RICH INTERNET APPLICATION TECHNOLOGIES	
3.1 The Key Technology: AJAX.....	15
3.1.1 Ajax Explained.....	15
3.1.2 Ajax in Action.....	16
3.2 Strictly Presentation Layer Based Technologies.....	19
3.3.1 Adobe Flex.....	19
3.3.2 Adobe Flash.....	19
3.3.3 Microsoft Silverlight.....	20
3.4 Frameworks and Mash-ups.....	19

CHAPTER IV. THE IMPLEMENTED PROJECT OF THIS THESIS.....	20
4.1 The Objective of this Service.....	20
4.2 The Services Provided.....	20
4.3 The Framework: SmartClient.....	21
CHAPTER V. A NEW FRONTIER FOR COMPUTER APPLICATIONS...	24
5.1 The User at the End of 2007.....	24
5.2 The Possibilities and the Future.....	25
REFERENCES.....	26

LIST OF FIGURES

Figure	Page
2.1 A Web 1.0 service (wiki main).....	9
2.2 A Page is Being Loaded.....	10
2.3 After the Page is Loaded.....	10
2.4 Gmail inbox- message open.....	12
2.5 Gmail inbox- message appended.....	12
2.6 Gmail inbox- pop-up reply.....	13
2.7 Gmail inbox- pop-up menu.....	14
3.1 Source: http://cristian.nexcess.net/ajax/quickstart/	17
4.1 MyWebTop- Sign-up form with calendar mashup.....	23
4.2 MyWebTop- with active Snaps.....	23
4.3 MyWebTop- the rich text editor.....	24

CHAPTER I

INTRODUCTION

1.1 Overview of this Paper

This paper will encompass discussions about the important technologies that are used as the building blocks for Rich Internet Applications. The detailed analysis of the technologies, their pros and cons, the classification of the applications that are being created with these technologies etcetera will be covered. One thing that must be mentioned here is that- this thesis was an *implementation* based thesis. So, the tools that were used on the project will be discussed in detail, along with what the project is all about- its structure, system, specs and limitations.

At the end of the discussions and analysis- we'll look at the change that's taking place very rapidly in application development and online services. We'll delve into the possibilities that are reaching for us, the new legion of application that might change the whole notion about the classification of application- local, online alike.

1.2 The Title Explained

The title is pretty much self explanatory- "Study and Implementation of Various Rich Internet Application Technologies and its Beckoning Possibilities". This thesis is about learning various technologies and tools that are used to build Rich Internet Applications- the only jargon that might just need explanation. Which is why a whole chapter (see Chapter II: RICH INTERNET APPLICATION) is dedicated on the matter. So, the title depicts that the thesis is all about Rich Internet Applications, its underlying technology and its possibilities.

CHAPTER II

RICH INTERNET APPLICATION

2.1 The Internet

The internet was originally designed for simply transporting information and documents. As technology has evolved- the World Wide Web has changed its direction in a variety of ways. It has definitely changed the way we live our lives. It has revolutionized the greater part of the world of business and the *whole* part of communication. It has connected millions and millions computers all around the world, with blazing speed, terabytes and terabytes of information is being passed every day. With the dusk now approaching for the first decade of the new millennium- we see the newest web-sites with all the blazing colors, slick animations, great presentation and usability. We see people selling their glasses or a car at eBay, looking at where to go in the summer at travelspec.com or just reading the news paper online. Starting from arranging a meeting- to applying for jobs- is done with e-mails. People don't write journals now- they just update their blogs. In the past fifteen years- it started with e-mails and posting information on the web, but now- a whole new plethora of services are presented all over the WWW. This change has given us the chance to differentiate between two stages of web application- the Web 1.0 and then- Web 2.0.

2.2 Comparative Analysis: Web 1.0 vs. Web 2.0

2.2.1. Web 1.0 Defined

As defined in webopedia.com-

“A system of Internet servers that support specially formatted documents. The documents are formatted in a markup language called HTML (*Hyper Text Markup Language*) that supports links to other documents, as well as graphics, audio, and video files. This means you can jump from one document to another simply by clicking on hot spots. Not all Internet servers are part of the World Wide Web.

There are several applications called Web browsers that make it easy to access the World Wide Web; Two of the most popular being Netscape Navigator and Microsoft's Internet Explorer.”

As we can see from the definition- Web 1.0 is basically what we used to be able to achieve through the internet ten years ago. The World Wide Web is a mesh, a very intricate and huge web of billions of HTML pages strung together. We, as users, would simply hop from one page to another, download/upload files. Let's look at an example of a basic, old fashioned, web service- wikipedia.

Yes, it is fairly new compared to the time specified on the previous paragraph- but the technology used in this service is Web 1.0. Now, a basic page from wikipedia might look something like this (see Fig.01) →

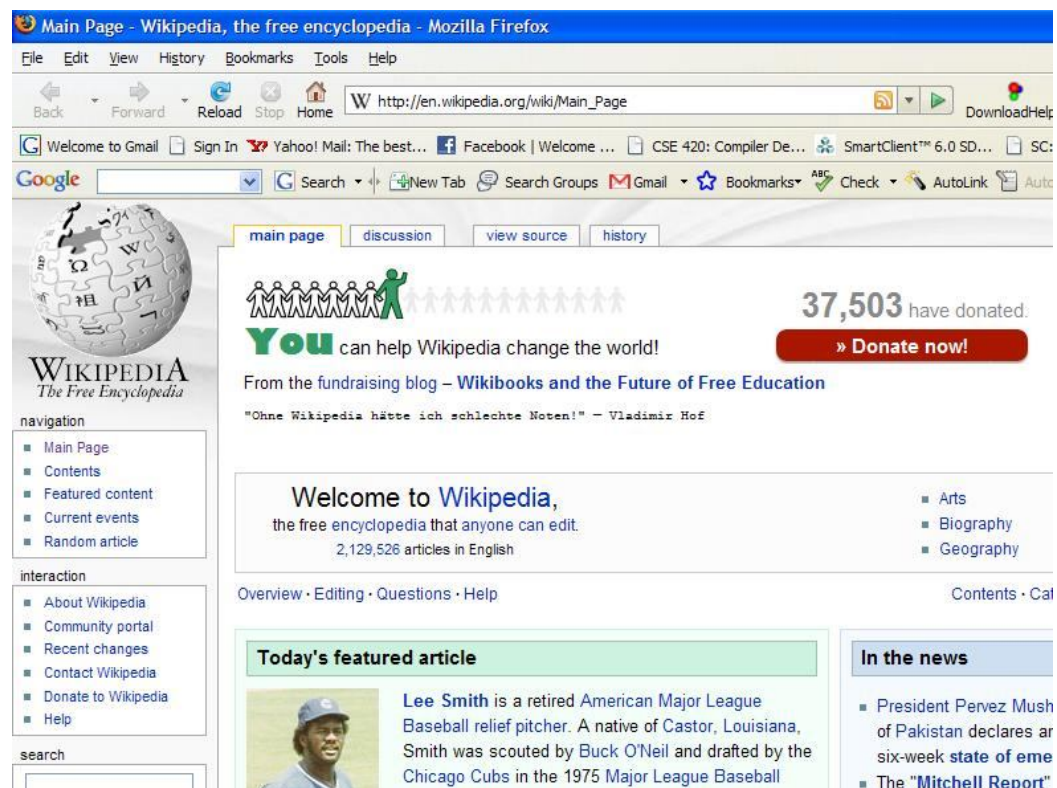


Fig. 2.1. A Web 1.0 service (wiki main).

Here, we see a basic HTML page that has hyperlinks all over it. Whenever you click on any one of the blue highlighted text- the browser will literally fetch the HTML page linked to that text from a remote server, and then decode and load it for you. It is actually readable, static data that is being loaded, reloaded, transferred and presented. So, if we click on a link of this page, essentially, another page will load (see Fig. 2.2 & 2.3) →

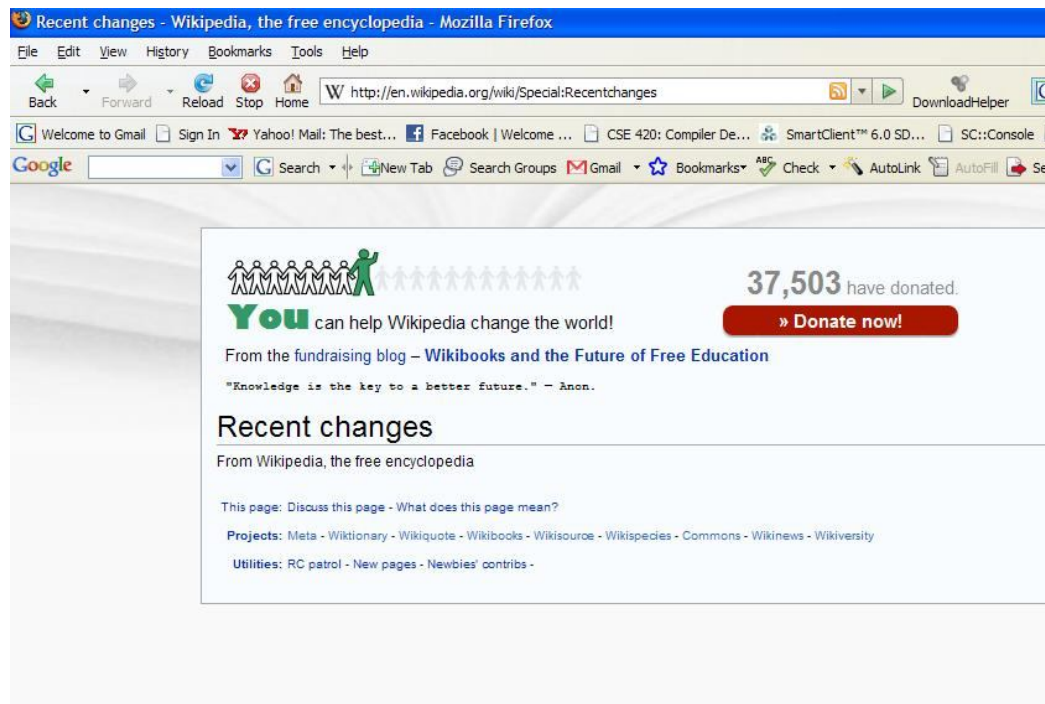


Fig. 2.2. A page is being loaded.

As the browser is decoding the received data (an HTML file), it starts decoding the embedded data (basically text and images) and presents it to the user. In Fig. 2.2- the page is being loaded, and it is loaded in Fig 2.3.



Fig. 2.3. After the page is loaded.

We can also say that, the Web 1.0 version web-sites or services are read-only. The user can only input keywords for search, login name, passwords etcetera- but can never change any *items* of the page. For example- the users won't be able to change the position of the image shown on the page, the color of the background- in all- users are unable to *customize* a web page. That's why we can say, that the web services that are categorized as Web 1.0, tend to have these traits-

1. read only
2. deliberate
3. static
4. not customizable

2.2.2 Introducing Web 2.0

During the last few years of the previous century, websites or web services have been changing. The technology in which the Internet was built upon- lacked interactivity with the users, unlike desktop applications. That has change in the past decade as innovators have developed richer user experience. Due to a break-through discovery of the technology called- Ajax, we can now deliver sophisticated, interactive, graphically alluring and responsive web pages to the users. Since these web services act just like desktop applications- they have been given the name- Rich Internet Applications- the members of Web 2.0. Before getting into the details of the technology, let us throw some more light upon distinguishing between a normal website and a Rich Internet Application.

RIA (Rich Internet Application) technologies not only revolutionized the user experience, it has empowered online business and development and took them to a whole new level. We'll look at one revolutionary web-service that was one of the pioneers in creating a web-mail with RIA technologies- Google. The google.mail.com service, famously known as Gmail, is a great example to demonstrate a rich web-client experience.

If we look at Fig. 2.4, we see as open message inside the Gmail inbox→

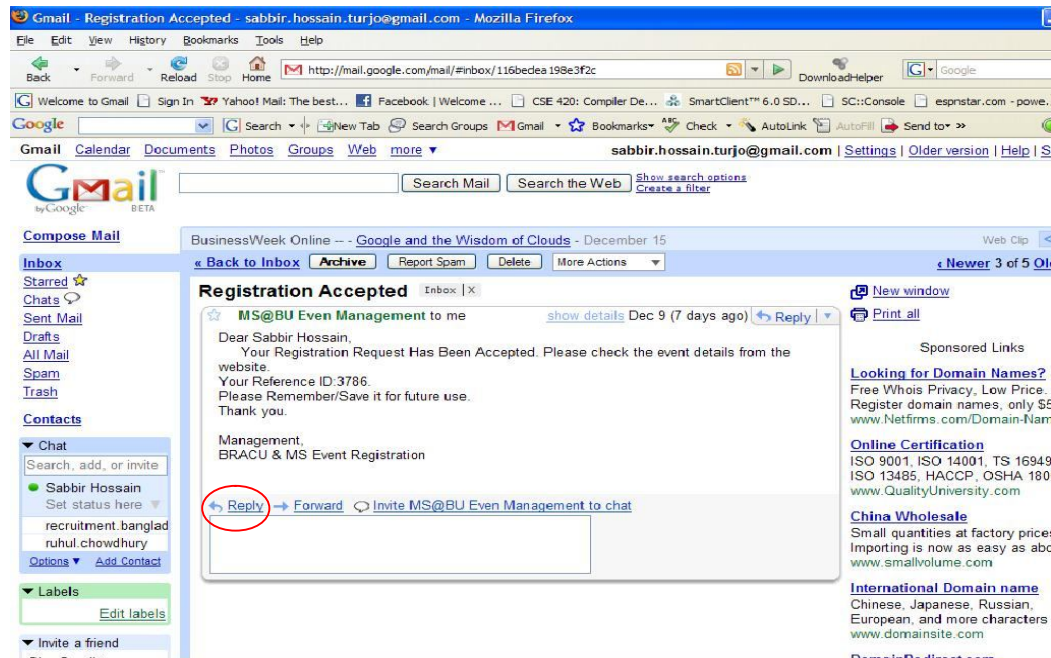


Fig. 2.4. Gmail inbox- message open.

Here, we see similarity with the previous example- hyperlinks all over the page. But, some of these links respond in a different way. Suppose, we click on the reply button. In a Web 1.0 theory- the current page will be discarded and a new page will have to be loaded. But in this case, a black message box is *appended* under the link reply (see Fig. 2.5 below) →

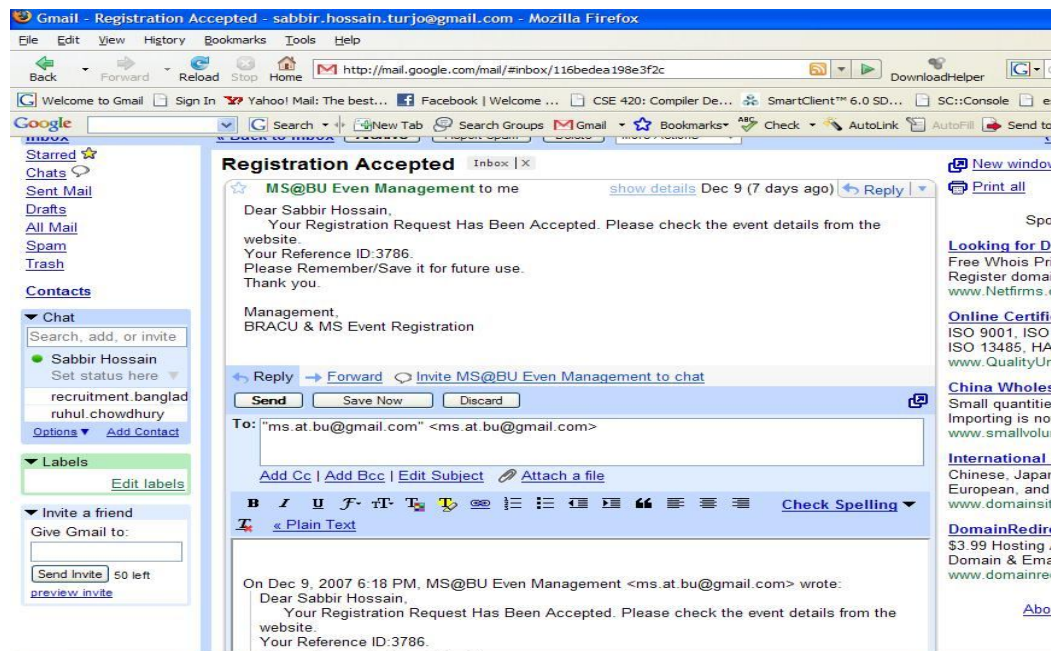


Fig. 2.5. Gmail inbox- message appended.

What happened in this case? The answer is pretty simple. Whenever the user clicks on the “Reply” link, data from the server is fetched (HTML data), decoded and added to the current page. But that is not the actual magic! What lies beneath this, is that this request to the server for data is done *asynchronously*. The word asynchronous means “not in synch, or not synchronized”. While the browser is transferring data with the server, the user will be able to continue her work, and the program will respond to each request- because the browser does not wait around for a single request's response. It goes back to its job and works on it only when a reply arrives. With this little theory in mind, we can do a bunch of stuff. For example- there's another reply button in the page →

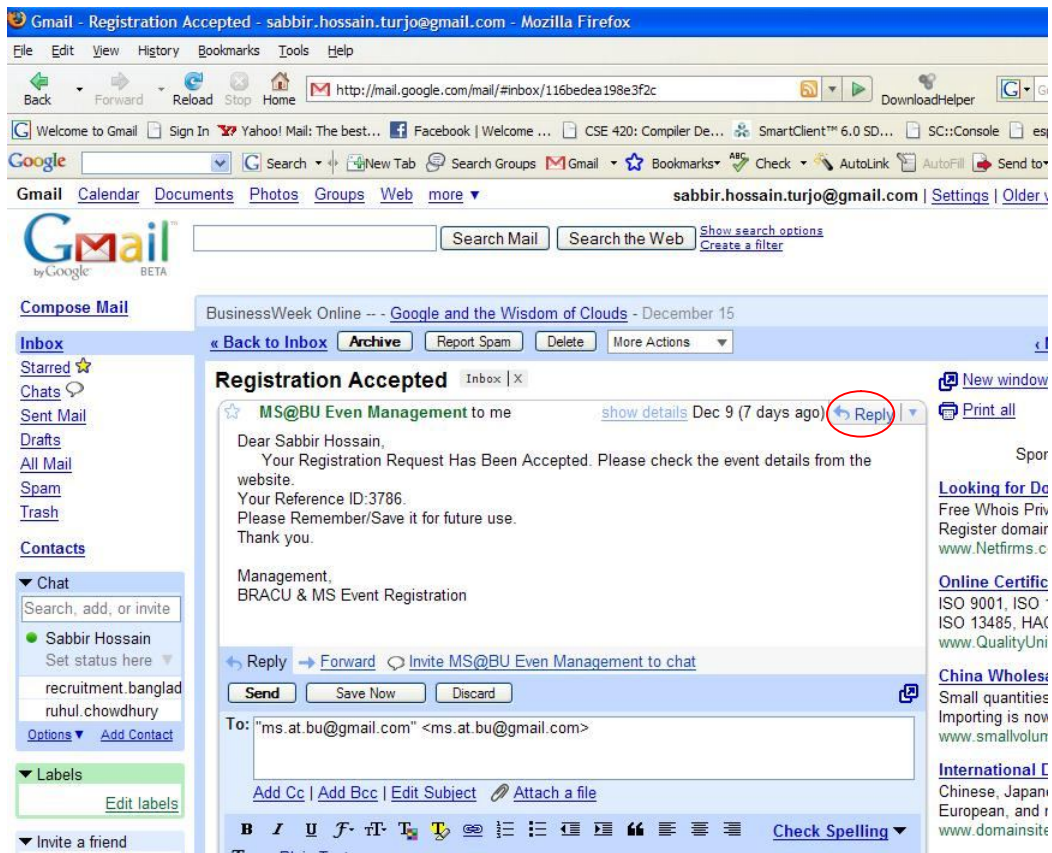


Fig. 2.6. Gmail inbox- pop-up reply.

If the user clicks this button, a pop-up menu appears (see Fig.2.6). The same actions are taken by the browser, but this time, the user is getting a different response- it is letting her to choose from given options.

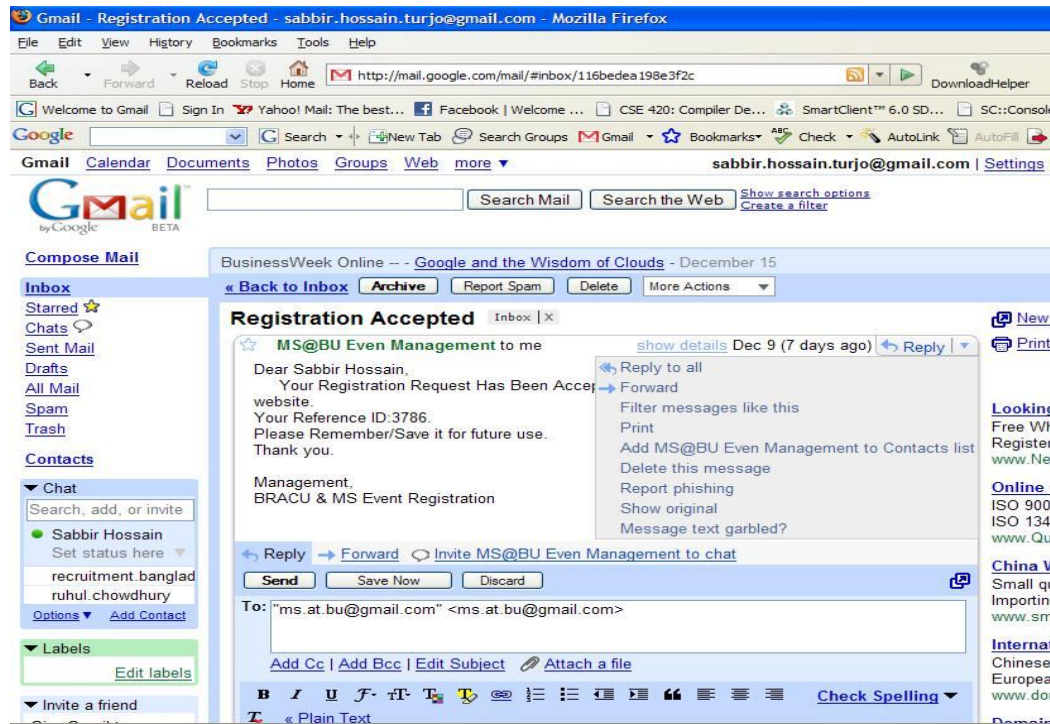


Fig. 2.7. Gmail inbox- popped reply menu.

So, by using this technology, we can give websites a more *desktop* application feel. By desktop application feel, I mean- quick, responsive user interactions through pop-up menus, message boxes, alerts, actions- all done while staying at the same web-page- without the page being reloaded. There are also hundreds and hundreds of websites where the user can drag and drop items like images, icons, small windows etcetera.

In the end, we can say that- Web 2.0 is the class of web services that have these following traits →

1. highly responsive
2. interactive
3. dynamic
4. writeable
5. customizable

CHAPTER III

RIA TECHNOLOGIES

3.1 The Key Technology: AJAX

3.1.1 AJAX Explained:

Ajax is- Asynchronous JavaScript And XML. It is based on related web development techniques like XML and JavaScript. Since a brief lookup on wikipedia or google will give the reader zillions of pages on AJAX- let's not talk too much about AJAX- just know what it is and how it actually works.

Using AJAX enables web pages communicate with the server “behind the scene”. The web page transfers request and replies- so that the entire web page does not have to be reloaded each time a request is made. Thus, AJAX is asynchronous in the sense that extra data is requested from the server and loaded in the background without interfering with the display and behavior of the existing page. JavaScript is the scripting language in which AJAX function calls are usually made. Data is retrieved using the XMLHttpRequest object that is available to scripting languages run in modern browsers. There is, however, no requirement that the asynchronous content be formatted in XML. But it is the default format for transferring requests and replies. Since it is written in JavaScript, AJAX is platform independent.

So, what are the possibilities that can become reality by using this technique? Well, the list is never ending! Controls such as trees, menus, data tables, rich text editors, calendars, and progress bars allow for better user interaction and interaction with HTML pages, generally without requiring the user to reload the page. HTML pages may poll data from a server for up-to-date data such as scores, stock quotes, weather, or application-specific data. A client may use Ajax techniques to get a set of current data without reloading a full page. Form data such as user IDs, serial numbers, postal codes, or even special coupon codes that require server-side validation can be validated in a form before the user submits a form. A specific portion of form data such as an email address, name, or city name may be auto-completed as the user types. Ajax techniques can be made to create single-page marshaled applications that look and feel much like a desktop application

3.1.2 AJAX in Action:

Now let's look at an AJAX driven web service closely, so that we can actually track down the path which a single data fetch and presentation would require.

Our example application would have the basic HTML file that would require the user to input the username and password, named 'index.html'. A JavaScript file named 'checkinfo.js' that is loaded on the client along with 'index.html'. 'checkinfo.php' is the php file on the server side that gets called by the JavaScript code in checkinfo.js. This web service, driven by AJAX, is described with the help of this diagram (see references) →

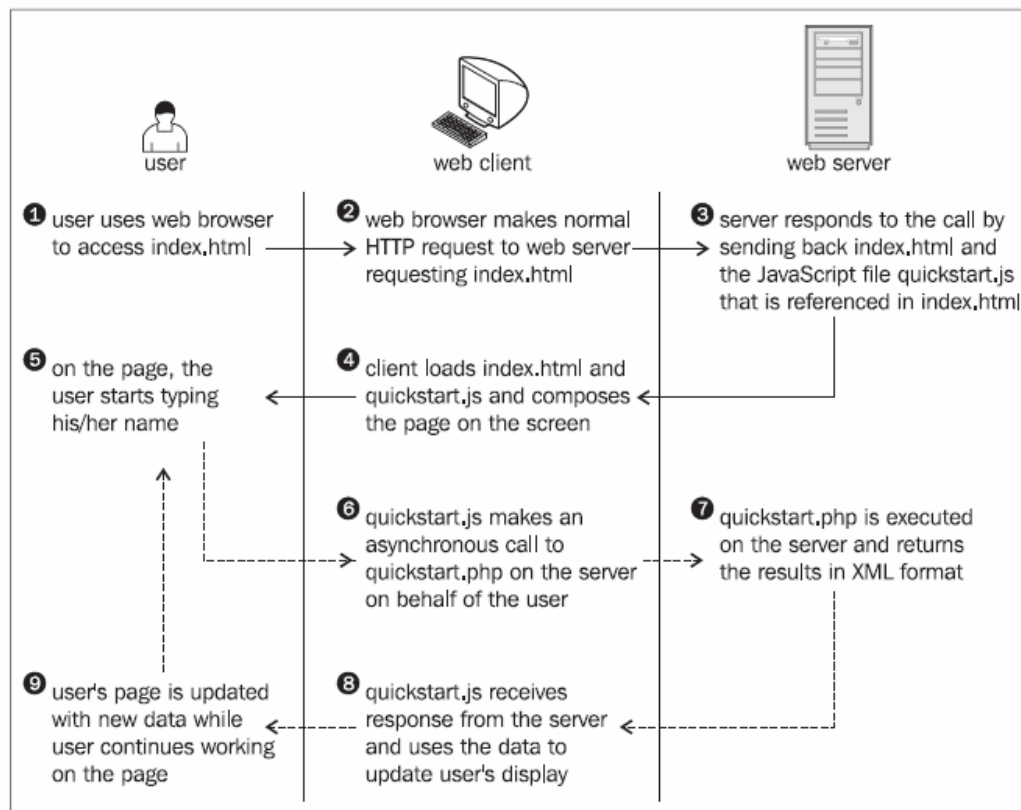


Figure 1.7: The Diagram Explaining the Inner Works of Your Quickstart Application

Fig. 3.1. Source: "<http://cristian.nexcess.net/ajax/quickstart/>"

3.2 Strictly Presentation Layer Based Technologies

3.2.1 Adobe Flex

Flex is actually is an SDK (Software Development Kit) and an IDE to support the development and deployment of cross platform, rich Internet applications based on Adobe's Macromedia Flash platform.

The goal of Flex is to allow web apps developers to quickly and easily build rich Internet applications. In a multi-tiered model, flex applications lie on the presentation tier. Flex was initially released as a J2EE application or JSP tag library that compiles MXML and ActionScript on-the-fly into Flash applications. Later versions of Flex support the creation of static files that are compiled at authoring time and can be deployed online without the need for a server license.

The most impeccable part of flex is that it helps integration of heavy animation, vector graphics and other rich UI components "on-the-fly". Meaning, as the Flex code is inserted into the *.swf file (the Flash binary file), the browser- upon loading the flash file- generates the flex components as it load the whole file.

Flex uses ActionScript and ActionScript 3.0 as the language (Flash ActionScripts) to code the UI components.

3.2.2 Adobe Flash

Originally released by Macromedia, Inc. in 1996 as a Web animation creation tool and delivery system, Flash has grown into a huge part of the Web development, design, and user experiences. The proprietary but free to download and mostly cross-platform Flash Player is both a standalone offline application and a Web browser plug-in that plays Flash files. The Flash file format encompasses vector and raster animation data; and logic in the form of a JavaScript-like, ECMAScript-based programming language called ActionScript. Flash Professional is a graphical integrated development environment (IDE) designed as both a drawing and animation program, and an ActionScript programming tool.

Flash Professional was originally intended to be a standalone program for creating programs and animations for the Flash Player, but was eventually made to work interoperably with Macromedia's other products -- the Dreamweaver HTML IDE, the Fireworks Web graphics creation program, and the FreeHand vector drawing tool. In their later incarnations, each program was capable of creating entire Web sites based on content you created -- be they drawings, animations, applications, storyboards, or hand-coded HTML.

Macromedia experimented with bundling the four products in duo packages, eventually including them all in the Macromedia Studio suite.

In December 2005, Macromedia was bought out by Adobe Systems, Inc., and both companies' products were merged into a variety of slightly different application suites. Adobe killed off FreeHand in favor of its own Illustrator vector drawing program, but smartly kept Fireworks for creating Web graphics and entire graphics-oriented Web sites.

3.2.3 Microsoft Silverlight

An addition to my survey during the pre-thesis period because of running out of possible technologies under the .NET framework- Microsoft SilverLight is the newest technology to the .NET family.

Silverlight is actually a runtime for Rich Internet Applications. It provides animation, vector graphics and video playback capabilities of the existing WPF (Windows Presentation Foundation). It can be integrated in any .NET language- C#, VB plus supports AJAX, Ruby and Python.

Silverlight is designed in cohesion with XAML and is scriptable with JavaScript. The XAML portion actually marks up the vector graphics and animations. SilverLight makes it possible to dynamically load XML content that can be manipulated through a DOM interface, a technique that is consistent with conventional Ajax techniques. SilverLight exposes a Downloader object which can be used to download content, like scripts, media assets or other data, as is required by the application.

A Silverlight application starts by invoking the Silverlight control from the HTML page, which then loads up a XAML file. The XAML file contains a Canvas object, which acts as placeholder for other elements. Silverlight provides various geometrical primitives like lines, ellipses and other shapes, to elements like text, images, and media etc. The elements are properly positioned to achieve the desired layout. Any arbitrary shape can be created as well. These elements can be animated using Event triggers; some animation effects are predefined, others can be created as composite of the pre-defined effects. Events like keyboard or mouse movements can also raise Events which can be handled by custom scripts.

Silverlight applications can be written in any .NET programming language. As such, any development tools which can be used with .NET languages can work with Silverlight. Microsoft has positioned Expression Blend 2.0 for designing the UI of Silverlight applications.

3.3 Frameworks and mash-ups

Frameworks dedicated to be used exclusively for web development- Rich Web Applications- are emerging rapidly. A software framework essentially means a reusable design for a software system, that has a set of abstract classes, and there is a cohesive manner in which they are used. Meaning, a framework is dedicated towards a definite kind of software-making. So, a RIA framework means a framework that is used for developing RIAs easily. GWT- or the Google Web Toolkit is a great example.

GWT makes AJAX based software development easy because the people at Google have developed thousands of widgets and UI components in this open source toolkit that can be reused, edited and shared amongst others.

The main features of GWT are →

1. Overabundance of reusable UI components
2. Server Communication is done by serializable Java classes
3. Adds state to the browser's back button history, makes browser history more manageable
4. At the development state, the program runs in the JVM. But in production, the code is compiled into JavaScript
5. Supports IE, Firefox, Opera and Safari
6. Google Gears is the newest addition to GWT. It's more like Dekoh and Adobe's Apollo, meaning it integrates online web services to offline applications. In other words, it makes developers to make web based applications function even when offline.

Mash-up, is comparatively a new term. The term was originally used in music- meaning the mixture of two or more songs into one. The idea is quite the same for web-based mashups. A mashup in terms of the internet- is a web application that combines data and resource from different source into a single service. The source mentioned here actually means previously defined public interfaces or APIs (Application Programming Interface). There are three kinds of mashups-

1. Consumer mashups
2. Data mashups
3. Business mashups

The mostly common one is the consumer mashup, which can be best described by services like the Google Earth application. Consumer mashups combine data elements from multiple sources, hiding this behind the simple unified graphical interface.

CHAPTER IV

THE IMPLEMENTED PROJECT OF THIS THESIS

4.1 The Project: Objective and Overview

Since the thesis was implementation based- the main aspect of it was to implement a project. And since my topic was studying about RIA and its technologies- I decided to create a RIA by using some of the most important technologies. This gave me the opportunity to dive into the bottomless ocean of web-digging. The best thing about these technologies is that you see them on the web in action, and if you want to learn them, the best place is the web itself. Hundreds and hundreds of books, tutorials and all sorts of resources are available everywhere.

The project is titled- "MyWebTop". By looking at the name- I'm sure the reader gets a good idea about the service. Yes, it is a desktop on the web, or, a WebTop. This service allows the user to use her account to personalize her browsing, even more, text editing experience. The detail of the service is explained shortly. Since I've mentioned 'desktop' on the name of the project, it has to have some relevance to a desktop. The website actually looks and feels like a desktop. Without further ado, let's jump into the details about- MyWebTop.

4.2 The Services Provided

First and foremost- this project is actually a consumer mashup. I have taken various parts from my existing work, embedded them into a framework, added up widgets from other sources. The framework that I used is an open-source, platform independent SDK(Software Development Kit). Details of the framework will be right after this. The things that are provided as serviced in this project are →

1. User Management: new user sign-up, existing user sign-in, sign-up forms, validation etcetera all are done using AJAX, on the same page, by loading dynamic windows. The calendar date chooser embedded inside the sign-up form is a widget from another source

2. A “Start” menu: to give the website a *desktop view*, there’s a menu bar on the top of the page, which acts and interacts with the user like the start menu of Windows.
3. A Rich Text Editor: there’s also a rich text editor in the start menu, which creates editable HTML pages- where the user can type, stylize, format text.
4. Web SnapShots: the user will be able to create frames that hold their desired web page. These frames, called web snapshots, are totally resizable, draggable, droppable. And the user will be able to create as many snapshots as they want. The state and icons for these snaps are saved, so that all the user have to do after login is click on those icons, the corresponding web page will be loaded inside a frame.
5. Save edited HTML: the edited file that is created in the rich text document, is saved on the file section, and by clicking on the button will generate the HTML related to the edited text.

4.3 The Framework: SmartClient

SmartClient is the framework which I used to implement the primary GUI(Graphical User Interface). This is an open-sourced, platform independent framework that provides the user with an SDK, a full-fledged documentation and a plethora of classes and interfaces that the user can use to generate great animations, slick designs, smooth and responsive web applications. It has a zero-install client engine and client-server data-binding system. So, what is data-binding?

Data-binding is used by AJAX to represent JavaScript data with XML, so that the server-client understands each other. But, the SmartClient framework uses the JSON (JavaScript Object Notation) data-binding system. During the last month of my thesis- I decided to use JSON as the data-binding scheme. So using SmartClient, actually complemented that idea of mine. A few snapshots of the project is given →



Fig. 4.1. MyWebTop- Sign-up form with calendar mashup



Fig. 4.2. MyWebTop- with active Snaps.

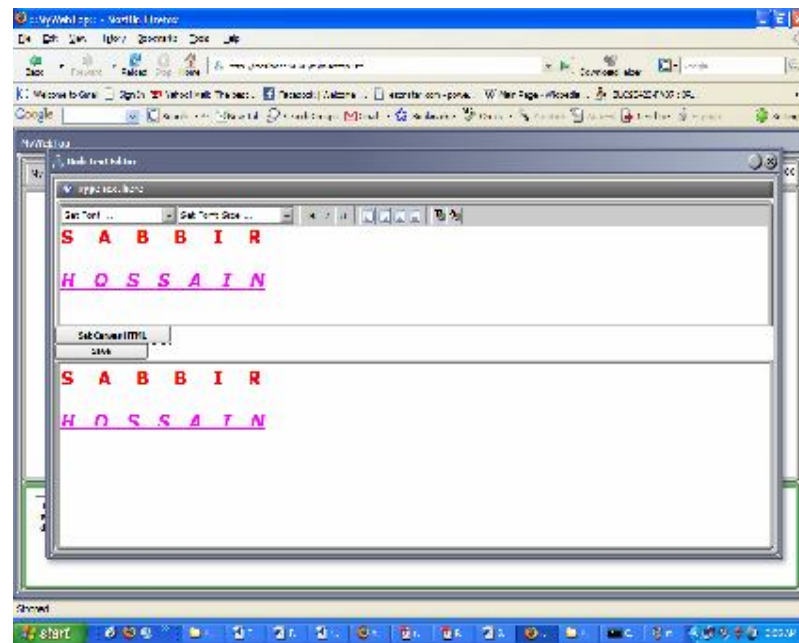


Fig. 4.3. MyWebTop- the rich text editor.

CHAPTER V

A NEW FRONTIER FOR COMPUTER APPLICATIONS

5.1 The User at the End of 2007

As I have mentioned earlier- due to the rapid change in technology, our lives are ever-changing. Human beings adapt to change. What depends on being the change welcoming and unwelcoming depends mostly on the credibility of the change- is it doing anything better? Is the change effective life positively?

From the start of this millennium, up until the end of 2007- a users' perspective towards the internet has changed drastically. People used to use the internet to exchange e-mails, get information about universities, books etcetera on the web. But now- in some areas in the UK and the US- people watch youtube more than regular TV! People don't upload file to a server or own web-site so that their kith and kin could get access to some photos- they go to webshots, flickr or picassa and upload entire albums. People use P2P (Peer-to-Peer) networks to share terabytes of data amongst the users. Large chunks of memory are mapped in servers by torrent files, so that people get access to more memory with ease.

Automatic feeds like RSS (Rich Site Summary), web-casts etcetera keep the user up-to-date without the user actually visiting the site. People share their thoughts in their Blogs (Binary Log), if its and important person to the user- she can sign-up on the blog and keep updated. One of the latest innovations are being done by companies like SmartClient- who'll provide the user with a framework, free if you want to develop for yourself- or purchase for rigorous developmental tasks- all these services available online. Travel agencies, airport web services have taken rich internet applications to higher levels. For example- if you go to serwin-williams.com, they will let you color, design the walls of a virtual house- *online!*

So, the user of the 21st century, do not look for a basic information take-away experience when she logs on to the internet. She looks for ways to make her life a better place to live in.

5.2 The Possibilities and the Future

The direction that World Wide Web, Internet and Internet technology is headed- possibilities are simply limitless. The world has definitely become a smaller place to live in- but the space that one can assess for working is bigger than ever. A person's ability to work- anywhere, everywhere is quite apparent. A simple example might just throw some light.

Suppose you have signed up for the google docs services. You constantly need most of the important documents, information, spreadsheets etcetera and you keep all your documents around with you in your laptop. Now, you are going out of the country and you couldn't possibly carry your laptop around. Since he has a google docs account, he can easily upload all his work in it, and access it from anywhere in the world that has an internet connection!

The computer started off as this humongous, singular building, with floors filled with machinery, coolers and what not? During the first 5-10 years, computers were *centralized*. There were quite a few terminals from one main-frame computer, which was shared by all the terminal users. Then technology soared to greater heights- things started to get compacted. Along came the personal computer, and a single user would get the power, attention and service of a single computer.

But, after the 1980s- the internet burst into the scene. It has rapidly engulfed the major portion of communication and information technology. By rapidly weaving the World Wide Web- are we not creating a humongous super computer? The WWW, is nothing but, millions of computer tied together. Just like the terminals from the main-frames! So, we are in a notion where we think that we are only progressing and that there's no time to look back. But we should sometimes stop the never-ending marathon called life, and think. Think how technology is heading backwards- at least metaphorically.

REFERENCES

- [1] SmartClient Home,
<http://www.smartclient.com>

- [2] Wikipedia, Mashup URL,
http://en.wikipedia.org/wiki/Mashup_%28web_application_hybrid%29

- [3] McLaughlin, Brett Mastering AJAX (Part 1-12),
<http://www.ibm.com/developerworks/web/library/wa-ajaxintro1.html>

- [4] Ajax in action,
<http://cristian.nexcess.net/ajax/quickstart/>

- [5] Domenig, Mark Rich Internet Application and Ajax- selecting the best product,
<http://www.javalobby.org/articles/ajax-ria-overview/>

- [6] Definitions,
http://www.webopedia.com/TERM/W/Web_1_point_0.html
http://www.webopedia.com/TERM/W/World_Wide_Web.html

- [7] O'Reilly, Tim What is Web 2.0?,
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

- [8] Barnett, Alex Alex Barnett's Blog,
<http://blogs.msdn.com/alexbarn/archive/2005/08/13/451282.aspx>