

Usability in the Web 2.0 World

by *S Ketharaman*

A couple of years ago, I had written an article on “Usability”, which was published in TECHNOLOGY EVALUATION CENTERS, a leading evaluator of enterprise application products. (You can download this article from <http://www.sketharaman.com/ARTICLE - Usability - v1.1.pdf>).

My recent experience with a few Web 2.0 applications has reinforced the belief I had expressed in my blog post in July this year (<http://www.sketharaman.com/blog/?p=15>), namely that, usability considerations are becoming even more critical in the current Web 2.0 world, in which Internet applications are coming increasingly into the mainstream and are being used by more and more people who are not IT-savvy.

Before we examine why usability plays such an important role in today’s world, let us have a quick look at typical Web 2.0 technologies and applications.

RSS, Blogs, AJAX, Podcasts, Wikis, Toolbars, Mashups – these are the basic building blocks of Web 2.0, the popular name given to the collection of technologies that underpin the current wave of Internet applications. Web 2.0 is characterized by user-generated content. Leaders in the Web 2.0 era focus on merely providing the platform whereas it is their users who provide most of the content. For example, **Jigsaw** is a business contact marketplace where users find, build and trade business contacts; **Wikipedia** is a free encyclopedia where users contribute articles on a wide range of topics; Flickr allows users to share, search and find photographs, and **YouTube** does the same thing with videos.

To illustrate the increasingly critical role of usability in the Web 2.0 world, let me use my recent experience with a Web 2.0 application.

Late August, I had written about Jigsaw on my blog “Talk of Many Things” (<http://www.sketharaman.com/blog/?p=21>). Many readers visited Jigsaw and found it very interesting. Some of the sales and marketing professionals amongst them even tried to trade a few contacts. But, a general comment I heard from most of them was how difficult they found it to share their contacts on Jigsaw.

I visited Jigsaw.com to get a first hand experience. Indeed, I found it quite tedious and time-consuming to upload contacts.

When I tried to upload my own contact into Jigsaw, I was somewhat disoriented to find myself getting recorded by Jigsaw as an employee of my company’s US subsidiary when I am actually an employee of the India-based parent company. Jigsaw seems to reference companies by the domain names in the contact person’s email address. As a result, a global corporation (e.g. i-flex solutions) using a single domain name (e.g. iflexsolutions.com) would find itself getting recorded in Jigsaw under a single name despite operating under different names in different countries. As a result, even though I am an employee of “i-flex solutions ltd.”, Jigsaw records me incorrectly as working for “i-flex solutions, inc.” I couldn’t find any way to change this.

If a business contact application is unable to capture a basic thing like the exact name of the contact’s employer, there’s surely a problem!

To upload multiple sample contacts into Jigsaw, I selected the option of comma-separated file (CSV). Once I did the upload, I got a message saying that the file upload was successful. I thought that was the end of it. But, no, I got a screen titled “Add Multiple Contacts” without any explanation about what I am required to do on that

screen. After starting with the add multiple contact option, you can imagine my confusion when I am brought back to the same command after completing a few steps in between – especially when there is no clue why I have been brought there or what I am expected to do next. In fact, on my first attempt to add contacts into Jigsaw a few days ago, I gave up at this stage.

I visited Jigsaw again after a few days only to understand the full range of usability issues involved in Web 2.0.

During my second attempt, when I reached the stage at which I had quit Jigsaw on the previous occasion, I decided to be persistent. After doing a bit of reading around on the website and connecting a few dots, I could figure out that I am actually being given the facility to decide which of my uploaded contacts I really wish to “put into the system”, that is, to make available to all users.

Well, when I started uploading a CSV file into Jigsaw containing a certain number of contacts, I was mentally prepared to share all those contacts with Jigsaw. To that extent, when I was presented with a choice of how many of the uploaded contacts I wanted to add into Jigsaw, my natural response was “of course, ALL”. However, try as I might, I just couldn’t find any button / hyperlink which would allow me to do that. Surprisingly, I found a prominent hyperlink to “Delete All Contacts” but none to add all contacts! I finally discovered that you could select all contacts by checking a small box on the header on the top. This box is not labeled, so it is quite easy to miss it out and go through the drudgery of checking each box on the left of every contact. Imagine how tedious and time-consuming it would be if you were trying to add 100 contacts at one time! 100 contacts would earn you anywhere from 500 to 1,500 points on Jigsaw, depending upon whether your contact’s employer already figures or is a brand-new entry in Jigsaw. You can imagine your plight when you learn that you need to

have at least 2,000 points in your account before you can start hoping to make money by selling them.

After getting past this screen, I came across another screen in which there was a red asterisk * marked against the address field “State/Province”, which was showing up as blank. I’d imagine that the red asterisk meant that this is a required / mandatory field. Funny thing is, number one, my CSV file included the state, so I was surprised why it didn’t show up here; and, number two, this field is dimmed and not editable. So, even if I wanted to enter the state name again, I was not allowed to do so!

There are many more such usability issues in Jigsaw. A detailed coverage of the same is neither the purpose of this article nor within its scope. To cut a long story short, I faced a few more problems, but finally managed to overcome most of them and completed the process of adding multiple contacts into Jigsaw.

From this example, you can easily appreciate how usability assumes additional importance in a Web 2.0 website that depends upon content shared by users. In traditional websites, content is uploaded by employees, so even a crude data upload script would suffice. Whereas, in Web 2.0, end users are uploading the content, so the process by which they do it must be made extremely usable. Otherwise, users will give up and the website will lack content.

In Web 1.0, content was king.

In Web 2.0, content provider is king and excellent usability is a key need of the content provider (i.e. the user) who is often not an IT-savvy person.

Web toolbars are another important Web 2.0 technology element. Many small and medium companies and leading not-for-profit organizations like Greenpeace have already started using web toolbars to enhance engagement with their online

audience. As I had written in my blog post in August (<http://www.sketharaman.com/blog/?p=19>), web toolbars have the power to draw a diverse spread of audience to your websites. Many actual users of web toolbars including Greenpeace have reported significant increase in their website traffic after distributing web toolbars. Please visit the testimonials page of Conduit, a leading provider of web toolbars (<http://www.conduit.com/Benefits/Testimonials.aspx>), for a few user comments. You can easily expect much of this traffic to comprise of non IT-savvy people who otherwise had no intention to visit the website but were enticed to land up there by the interesting RSS feed on their web toolbar notifying, for example, an increase in term deposit interest rates offered by their bank or a reduction in home lending rates offered by their mortgage provider.

Now, you can imagine the additional demands on usability when there is explosion in traffic, lot of which comprises of non IT-savvy users.

When poor usability could lead to irritation and fatigue even in the Web 1.0 world, imagine what it could do in the Web 2.0 world where there are more users, many of whom are not IT-savvy and most of them don't just visit the website to retrieve information but to actually provide the content on which the website runs! The results of poor usability in Web 2.0 applications would be catastrophic. Simply imagine what would happen if users found it too difficult to use the photo upload feature in a typical photograph-sharing Web 2.0 website like Flickr.com. The site would be absolutely empty of content. That Flickr contains a huge collection of photographs is testimony to the ease of use of its photo upload feature.

However, Flickr has gone beyond traditional PC-based uploads and has recently launched a facility whereby users can upload photos directly from their Nokia Nseries mobile

phones. In this context, usability demands and challenges take on different meaning. Screens, menus and search result pages will have to be compressed into the smaller display area of a mobile phone. While the application itself hasn't become any simpler, it would be necessary to provide highly condensed help screens and dialog boxes suitable for a mobile phone's reduced form factor. Text entry will need to be minimized because it can be quite painful to enter text into a mobile phone's cramped keypad, especially if the user is traveling in a crowded public transport bus, walking under the hot sun or over slippery snow!

When you consider these aspects, you can begin to understand how important it is to incorporate the next level of usability features into Web 2.0 applications of today and tomorrow.

The good news is that rapid strides in Web technology in the last couple of years are making enhanced usability very much achievable.

While web technologies have traditionally supported reasonably high degrees of usability, the very simplicity of applications built using them seemed to rule them out for building applications that have called for higher degree of responsiveness and richness. **AJAX** and **Adobe Flex 2** are two recent web technologies that hold tremendous promise for bridging this gap.

In the words of AJAX.org, the official AJAX special interest website,

AJAX stands for Asynchronous JavaScript And XML. AJAX is a web development technique used for creating interactive web applications. AJAX is the logical next step in the services-oriented architecture revolution. With AJAX, user interfaces from within the browser can use web services as their data source to store and retrieve information. Asynchronous JavaScript and XML is

not a technology in itself, but is a term that describes a "new" approach to using a number of existing technologies together, including: HTML or XHTML, Cascading Style Sheets, JavaScript, The Document Object Model, XML, XSLT, and the XMLHttpRequest object. When these technologies are combined in the AJAX model, web applications are able to make quick, incremental updates to the user interface without reloading the entire browser page. This makes the application faster and more responsive to user actions.

AJAX applications can change in real time. You no longer have to wait for 5-10 seconds to reload a web page each time you click somewhere. Just one part of the page can get refreshed with new information fetched from the server without incurring the overheads of reloading the entire page. AJAX applications also permit users to drag boxes around and re-organize a web page without clearing it. As a result, AJAX is ideal for web applications that demand a high degree of usability and responsiveness.

With Flex 2, Adobe promises to go beyond AJAX in building interactive and rich Internet applications (RIAs). On its website (<http://www.adobe.com/products/flex/>), Adobe states,

Adobe Flex 2 revolutionizes the way people interact with the web.

Adobe® Flex™ 2 is the most complete, powerful application development solution for creating and delivering cross-platform **rich Internet applications** (RIAs) within the enterprise and across the web. It enables the creation of expressive and interactive web applications that can reach virtually anyone on any platform. Enterprises can use Flex to quickly build and deploy applications that improve the user experience, boost the bottom line, and analyze

data to enable better business decisions.

Although relatively new, both AJAX and Flex 2 already boast of many applications in areas like financial trading and settlement, account provisioning and enrollment, call center and CRM, order processing, field sales and service, product configuration, and dashboards. A quick glance at some of these applications on <http://www.adobe.com/devnet/flex/?tab:samples=1> will give readers an idea of the immensely powerful usability features supported by these technologies.

Apart from such new technologies, change in the traditional programmer mindset can also bring about measurable improvement in usability.

Most programmers lack empathy for an average user and write software for IT-savvy users. They often tend to assume that their users will have help on hand while using their applications. As a result, they underestimate the importance of user-friendly GUIs, simple navigation schemes and self-explanatory instructions and error messages.

How flawed this assumption is! In the Web 2.0 world, increasing number of users are non IT-savvy and lack the time and inclination to read help screens or user manuals. As corporations increasingly cut costs, helpdesks are dwindling in size. There is increased emphasis on developing browser-based self-service applications not only for external users accessing them via Internet but even for internal applications (e.g. Employee portals for leave applications, expense claims, etc.) that are meant solely for Intranet-based employees.

Let alone better GUIs and improved navigation schemes, even a little self-explanatory text placed at the right place on the screen can go a long way to improve usability of applications. For example, in the Jigsaw example narrated earlier, once the

user has successfully uploaded her CSV file into Jigsaw, it would help if she could be presented with the following statement at the top of the next screen: "While your file upload with all your contacts is successful, you are now being given the chance to be selective about which contacts you would like to make public to other Jigsaw users. Click here to select ALL contacts OR select only the ones you want, from the list given below." To quote another example, most corporate remote email applications expect their users to provide not only their usernames but also their network domain names while they are trying to access webmail, even though the screen specifies only username. Whereas, over a Virtual Private Network (VPN), users are required to skip the network domain name in order to gain access to their emails. It would help if the webpage clearly shows a sample entry just beside the username field.

Some programmers I recently spoke with told me that there is not much progress happening on frontend technology, so there's no fun in focusing there.



About the Author

With a B.Tech from IIT Bombay and a Master's in Marketing Management from Jamnalal Bajaj Bombay, **S. Ketharaman** has over 20 years of work experience in the IT industry.

Through his career, **S. Ketharaman** has worked in various roles in sales, marketing, delivery, program management and general management and across a wide range of markets like India, USA, UK, Germany, Switzerland, Singapore, Malaysia, Thailand and Middle East. Apart from solid experience in SAP, PeopleSoft and Ramco ERPs, **S. Ketharaman** has strong expertise in banking and payments solutions like Internet Banking, Payment Hubs, Cards, EBPP and Remittance. Earlier

in his career, he acquired experience in hardware and networking products including Unix-based super-mini computers, Wintel servers, desktops, notebooks, WAN and IBM mainframe gateways. **S. Ketharaman** began his career in the engineering industry, where he worked for two years before moving into the IT industry.

Over the years, **S. Ketharaman** has developed frameworks and models in areas such as identification of differentiators, price forecasting and innovative service delivery. He has also authored many articles on topics like Web 2.0, software usability, ROI of ERP and ERP customization; these articles have been published in leading publications in India and USA. He also runs a blog called TalkOfManyThings.com, where he writes on a variety of topics including sales and marketing, technology, globalization amidst cultural differences, gadgets and gizmos, and so on.

Based out of London, UK, **S. Ketharaman** currently works for *i-flex solutions* and heads its payments business with a large UK-based global bank. For more information, please visit www.sketharaman.com or e-mail info@sketharaman.com / s.ketharaman@iflexsolutions.com.

Views expressed in this article are personal.

How mistaken they are! Looks like they haven't heard of Web 2.0, AJAX, Flex 2 and the like. Nor have they realized that many large banks have recently commissioned multi-million dollar IT projects to revamp the frontends of many of their applications using AJAX with the goal of cutting down transaction processing times and costs.

I can go on with several examples, but the point is Web 2.0 applications are making increased usability demands, which can fortunately be fulfilled by a combination of latest technologies like AJAX and greater empathy for the user from the software development community. As Web 2.0 development platforms like *Ruby on Rails* and *Nexaweb* reach the mainstream of web application development, and as business applications based on Web 2.0 technologies reach critical mass, I shall cover specific usability related features supported in each of these platforms, and their relative merits and demerits. Watch this space!