What You See is What You Get: The Rise of the Web as a Platform

Abstract

The rapid exchange of data on an interconnected world-wide fabric called the Internet and a standard language of presenting and navigating documents called HTML, have created the extremely successful platform of the Web. This nascent platform is popular since it allows easy access to information, but most importantly since it permits actions to be performed at a distance. Actions like mailing, trading, shopping, banking, government, business and even dating have, therefore, accrued an "e-" prefix. However, the limitations of the Web in rich programmable components and its simplistic viewing model, have pushed webpage designers to new levels of creativity in order to accommodate the requirements of using the Web as a sophisticated platform. Such machinations leave people with visual impairments in the cold since on the Web "What you See is What You Get", meaning that principles of good user-interface design have been put aside in favor of fast deployment and quick access by the users. Or, in other words, on the Web "If it looks ok, it is ok". This paper seeks to analyze the current user-interface issues facing the Web.

Introduction

Frustrated I read the e-mail a second time. Hidden deep down within a table which is itself within another table placed there for purely decorative purposes, is a link which when clicked will open a website which has 20 links and somewhere in the middle of all of them one should find a table in which one should type their credentials and then press onto a Web button in order just to start the process of enrolling into campus housing for my university; or so the e-mail reads. Of course the e-mail does not say all of these, but one can surely infer them from past experience. And after starting the process of enrolling, one has to also finish it, i.e. spend the better part of an hour reading webpage after webpage full of boxes, intermingled with text and find where exactly each piece of requested information should go, reading linearly item after item with a screen reader, of many times useless navigational links while waiting for each page to load. Indeed, welcome to the age of the Web. For if the past we had to talk to human beings and wait in lines to get our work done, now we have to talk to the Web and wait in line until a so called server decides to serve our pages in small piece after small piece, as much as it can handle. For, if in the past blind people had to put up with the burden of navigating the streets, now they have to successfully navigate the e-streets.

What has brought us to this point? A point when the cheap work of filling forms and reading shopping or price lists has not been assigned to a helpful and smiling employee of the establishment but has been designated to the customer itself. The mantra is "Please waste **your** time, not **our** time". "Want to find out more?" or "Want to find out anything at all?" they say "Visit our website!" "Want to apply to our university program?" they say, "Spend your time, not our time, your time and fill the forms on our website." "Want to find out when the power outage will be restored?" they say, "It will be announced

on our website!" Even though no electricity means that you will not be able to even read the website, nevertheless, it will be announced there.

All this reliance on the Web would have been warranted if it had been originally designed for the purpose. However, the Web is a mash of technologies hastily put together to serve the consumer. And strangely enough the consumer loves it. "Want to read your e-mail from anywhere?" "No problem, Use our enhanced Web-mail site." And even though "enhanced" does not mean "better than the desktop experience", nevertheless, the fact that you can "read your mail from anywhere" in itself makes it "enhanced" I suppose. Conference after conference one reads a great number of papers, which do not try to improve the design of applications or simplify user experiences or better invent new interactive controls, but which simply port an existing off-line or desktop technology to the Web. Want a Webified tree-view control, there might be a paper for that. Want how to draw 2d graphics on the Web, no worries, since some researcher has looked into that too. It seems, therefore, that user experience has been relegated to a last place, whilst the first places have been taken by ease of deployment and perhaps a more "forgiving" development process, since traditional compiler bugs do not exist on the Web where browsers vie to accept anything being thrown at them.

Whatever though our wishes or, to be exact my wishes, are, it seems that the Web is here to stay. We are indeed entering the era during which, if you do not exist on a search engine, you may just as well do not exist at all and so, in order to facilitate discoverability, privacy does not apply to anything that leaves your head or even in that case, if it is "a matter of national security". If living with Web applications is inevitable, the following sections describe what changes should be undertaken to make Web-pages more welcoming to people with visual impairments.

Removing Unnecessary Clutter, Or, Avoiding the "Link, Link, Link"

The more links a user has to click, the more likely that she will abandon the Web-site and try her luck somewhere else. Designers, in an effort to keep users happy and make them spend more time on their site, have the tendency of including every link imaginable or available on all the pages of that site. Thus, on all the pages of a site one will always find repeated links which are everywhere the same. They are known to some as navigational links or top-navigation links, etc. They are usually placed towards the top and bottom of each page, but can also appear everywhere and are not relevant to the current task at hand, but are there for "convenience".

These links, which include common names such as "Home", "Products", "Search", "Privacy", "Downloads", "Profile", etc, are extremely annoying to people who use a screen reader, since they increase the amount of text which must be read on the page as a whole by such a user. Their presence makes reaching the important parts of a Web-page slower and although some screen readers have a feature to skip them, such a feature should be activated by the user, is error-prone and does not make navigational links available to the user if they are ever required. All that a blind user hears in such a case is a series of "Link this" and "Link that" and "Link the other" from his screen reader. Many users are even known to have broken their cursor down key: So many times they had to press it in their efforts to skip all these links.

Contrast this to a desktop application, where the function of navigational links is performed by the menu and then one can easily understand the great degradation in user experience that navigational links cause. On the desktop, menu items are easily navigable in four directions by using the cursors, and not linearly like navigational links. They are quickly accessible through designated shortcut keys and one does not need to hunt for them. Most importantly, they are grouped into logical categories, like File, Edit, View etc, which are consistent across applications. Unlike navigational links which are different on every site, menu items are not part of the main window's contents and are, as it were, tacked to the side, or to be exact to the top, while the user is using the application. Given the above, navigational links should therefore be grouped, be removed from the page's contents and be placed in a menu easily reachable by shortcut keys from anywhere on any of the site's pages.

Stitching Related Web-Pages in a Collage, Or, Avoiding the "Next, Next, Next"

The fact that Web-pages take time to download and that in a Web-browser it is hard to open multiple windows from a single site and have them easily and visually associated with one another, have forced many designers to provide or to ask for information in a fragmented, piece by piece approach. For example, the search results on a search engine are displayed page by page and it is not a wonder, therefore, that most users seldom go passed the first page. For example, when filling-in an on-line form one has to continuously press "Next, Next, Next" at the bottom of every form page.

For visually impaired persons, the above model of presented fragmented information is frustrating since one loses focus every time the page is refreshed or advanced. If, for example, one is reading a list of search results and she presses the "Next" link, she has to move again to the top of the page and skip through any navigational links or other controls which are their, or, even other information unique to that page, in order to reach again the desirable position where the list of results should continue. Would not it have been better if the list of results were to scroll indefinitely? In fact, a special plug-in was created for that purpose but from what I know, its functions are limited to only sites which have been programmed into the plug-in.

Another example,, is the many on-line books or computer manuals, which are published in a fragmented way, by which each sub-section is its own page and again users have to continuously press "Next" or "Previous" to advance forward or move backward. Would not it be easier if one could search or use any navigation commands, such as "Jump to Next Heading", on the whole of the document, instead on a page by page basis? Would not it be simpler if all these pages were in a way linked or integrated like the book they represent in a page collage?

A proposed solution should have three folds, therefore. It should automatically advance the user to the next or previous page of text or form controls upon the user going passed the last "interesting" piece of text or form control, (excluding navigational links), and it should link all such related Web-pages in a collection called a collage. It should automatically place the user on the text or control which is similar in nature or related to the text or controls she was dealing with on the previous or next page before the page was automatically changed. For example, if the user was filling a form and navigation was

automatically moved to the previous page, it should find the last control on that page related to the form the user was filling in and give focus to that control. Thirdly, the solution should make discovery and navigation commands, such as "Find" or "Next/Previous Heading", be available globally on all the pages included in the collage.

Fixing the Navigational Model, Or, Avoiding the "Tab, Tab, Tab"

Currently, the only navigable places on a Web-page are the links and all the form elements, such as the button and the edit field. This means that the only controls which are reachable using tab are the above. However, there is no hierarchical grouping of the navigable elements or of the navigational keys.

In a desktop application there is a clear hierarchy between controls. For example, an application window might contain two panels (frames), one of them containing 3 toolbars, which in turn contain various buttons and edit boxes, and the second panel might contain a list-view control with various items including a picture per list-item and a rich edit field. In fact, the just described application might be an e-mail client. Navigating from panel 1 to panel 2 and back might be performed, for example, using F6, changing between toolbars might be done using ctrl+tab and navigating amongst their buttons using the cursors. Similarly, navigating and resizing the list-view might be done using the cursors and switching from it to the rich edit box and back might be possible using tab. In other words, a plethora of navigational keys at each level of the navigational hierarchy: F6 for large objects, ctrl+tab for smaller objects nested within the large ones, tab for even smaller items, etc, etc.

However, the information delivery nature of the Web as opposed to the action-oriented nature on the desktop, does not include such a hierarchy in navigation. In addition, the lack of various advanced form controls on the Web has forced designers to mimic the absent controls using a combination of text and other irrelevant controls. For example, a tree-view control might be mimicked using a combination of "+ and "-" form buttons and specially colored and indented text between them. For a sighted person who does not use a screen reader "What you see is what you get" and he might not be able to even tell the difference between the above fake control and the actual one, since they visually resemble in appearance and in functionality. However, a screen reader, lacking the abilities and the past experience of a real user, cannot guess what unavailable in HTML control a series of other smaller controls, of colored text and other effects try to represent.

The hierarchy of HTML elements, although present is in no way, however, related or apparent in the navigational model of the Web. For example, for purely programmatic and styling reasons, a div element in HTML might house a form in order for the designer to be able to easily control its font and color through CSS and not for navigational purposes. Also, a table might be used for the easy placement of other children items on the page. The table might have no relationship to the format of the data or elements housed in it. But even, if the hierarchy of HTML elements were to reflect the navigational paradigm of the specific page, there is no tool for the user to gain access to such a hierarchy. As we said above, the only navigational key is tab. Thus, the absence of multiple levels of navigational keys does not allow any navigational hierarchy to be exposed. The result is that many blind people are forced to press "Tab, Tab, Tab" many times until they arrived at their desired place on a page. And although other text-

based navigational methods are usually available, e.g. navigating by text heading, etc, they are dependent on the good-will of the page's author, represent and are thus tied to their visual representation and do not provide any hierarchy.

Further, the Web does not provide rich mechanisms of identifying which items or HTML elements are linked or are related. For example, a forum might provide a list of conversation threads. Each thread might have a list of posts by various people. In HTML one (and especially a screen reader) cannot tell that the assortment of links, text, headings, and buttons, for example, represent a list of forum threads or a list of posts within a thread. Whilst to a sighted person such hierarchical groupings and item separation, e.g. the separation of one post from another and the grouping of replies to a specific post together, are easily discernible through visual clues or differentiation on the page, a user using a screen reader can only "see" an assortment of links, text, buttons, etc, which need to be read as a whole and then grouped or separated in the user's mind. Such a serial mode of processing information hinders the fast assimilation of contextual data and renders the finding of information excruciatingly slow. In fact, to many blind people navigating the Web has become a daily struggle, as more and more employers turn to Web-based applications in order to save costs.

Given the above a solution should therefore: Hierarchically group and present information, i.e. if a forum site has say a search form and a list of forum threads, it should create a hierarchy in which at its top should be the search form and the list of threads and allow the user to drill down into the hierarchy, expanding or collapsing each of its "branches" as needed. It should provide a list of different shortcuts for quickly navigating each level of the produced hierarchy, e.g. ctrl+tab for the higher level, tab for the items in the lower level and the cursors for yet the lower level. On a forum site again, ctrl+tab would be used to navigate, for example, between the form used to search the forum site and the list a forum threads, tab would be used on the search form to navigate between the search entry field and the "Search" button and in the forum threads list between the list and controls such as the "New Post" button, etc, and the cursors would be used in the thread list itself to go from post to post. The solution should be able to separate amongst each logical item at each hierarchical level and allow the quick navigation amongst such items. For example, on our forum site, our solution should be able to separate each post from another and be able to give the user the option of manipulating each post as a whole, i.e. the name of the poster, the post's date, its text and any associated action links, should be presented as a single post item for the purposes of navigation and the user should be able to discover each detail of the post item, e.g. the name of the poster, independently. For another example, on a shopping site, the solution should be able to identify and separate each listing on its own and treat it as an independent unit of information which collects details such as e.g. the name of the item, its price and its rating. The user should then be able to quickly navigate using a single key from listing to listing and review each of its details independently and if and when she wishes, i.e. I might want to quickly reach the 5th listing on the shop's site in the category PCs and read only its price.

Conclusion

The Web is evolving at an alarming rate. Although originally created as an information dissemination medium, the Web has grown to include action-based applications which take information from the user,

process it and produce results. Results which are even more important than any result that an off-line or desktop application might produce, including on-line banking, shopping and filing taxes. Unless, a fast and simple model of navigation is created to handle such Web-applications for the people who use screen readers, the Web would become harder and harder to use. This paper has proposed the removal of Web clutter, the automatic linking of related Web-pages into a collage and the automatic grouping and separation of logical items on Web-pages.