

An Investigation of Usability Issues in AJAX based Web Sites

Chris Pilgrim

Faculty of Information and Communication Technologies
Swinburne University of Technology
Hawthorn, Vic, Australia
cpilgrim@swin.edu.au

Abstract

Ajax, as one of the technological pillars of Web 2.0, has revolutionized the way that users access content and interact with each other on the Web. Unfortunately, many developers appear to be inspired by what is technologically possible through Ajax disregarding good design practice and fundamental usability theories. The key usability challenges of Ajax have been noted in the research literature with some technical solutions and design advice available on developer forums. What is unclear is how commercial Ajax developers respond to these issues. This paper presents the results of an empirical study of four commercial web sites that utilize Ajax technologies. The study investigated two usability issues in Ajax with the results contrasted in relation to the general usability principles of consistency, learnability and feedback. The results of the study found inconsistencies in how the sites managed the usability issues and demonstrated that combinations of the issues have a detrimental effect on user performance and satisfaction. The findings also suggest that developers may not be consistently responding to the available advice and guidelines. The paper concludes with several recommendations for Ajax developers to improve the usability of their Web applications.

Keywords: Ajax, usability, world-wide web.

1 Introduction

The World Wide Web has evolved in both size and uses well beyond the initial conceptions of its creators. The rapid growth in the size of the Web has driven the need for innovation in interface technologies to support users in navigating and interacting with the increasing amount of diverse and rich information. The technological innovations over the past decade have strived to provide users with a supportive interface to access information on the Web with ease including new Web 2.0 models of interaction that allow users to interact, contribute, collaborate and communicate. However, with this innovation there has been an unintended consequence of increased and additional complexity for users. The new models of interaction have in some cases required a shift

in the paradigm of how users expect the Web to behave. Users have had to change their view of the Web, from that as a vehicle for viewing content, to a view where the Web becomes a platform by which applications and services are delivered (Dix & Cowen, 2007). This paradigm shift breaks one of the fundamental principles of the architecture of the World Wide Web as having the unit of the 'page' (Berners-Lee, 1989).

One of the underlying technologies behind this evolution is Ajax. Ajax is now regarded as one of the technological pillars of Web 2.0 (Ankolekar et al, 2007) by providing the basis on which the Web can be regarded as a 'platform' for the delivery of services and applications that promote "openness, community and interaction" (Millard & Ross, 2006). Whilst the world has benefited from the evolution of the size and uses of the Web, the rush to embrace innovation has resulted in many developers to overlook well-established principles of good design and usability (Nielsen, 2007).

Ajax has several usability issues that have been reported in published research. In response to these issues some developer forums have provided design guidelines and technical solutions that Ajax developers could employ to alleviate any undesirable usability effects in their Web applications. What is unclear is whether commercial Ajax developers respond to these issues. This paper presents the findings of an empirical investigation into a set of Ajax enabled commercial websites to determine how web developers are responding to these usability challenges. The first section of the paper discusses the features and benefits of Ajax technologies. Some general heuristics for usable computer systems are presented and the specific usability challenges of Ajax are discussed. The methodology and results of the study are presented with the final section presenting a discussion and several recommendations for developers.

2 AJAX – Features and Benefits

The term Ajax has been attributed to Garret (2005) who coined it as an acronym for "Asynchronous Javascript and XML". In his essay Garrett described the five key characteristics of Ajax based applications as:

- a user interface constructed with open standards such as the Dynamic Hypertext Markup Language and Cascading Stylesheets (CSS);
- a dynamic, interactive user experience enabled by the Document Object Model (DOM);

- data exchange and transformation using the Extensible Markup Language (XML) and Extensible Stylesheet Language Transformations (XSLT);
- asynchronous client/server communication via XMLHttpRequest; and
- JavaScript that binds all the components together.

Various techniques of combining these technologies to create richer interactive experiences for Web users preceded Garrett's essay. Dynamic HTML combined static HTML with a scripting language such as JavaScript and a Document Object Model to dynamically manipulate CSS attributes of features on Web pages. Remote scripting, hidden frames and iFrames have also allowed Web developers to dynamically control content of components of Web pages. Ajax brings together these approaches and techniques into a quasi-standard that is now supported through various integrated development environments and application platforms such as Microsoft Atlas, Dojo and Ruby on Rails. The recognition of the benefits of Ajax technologies increased due to the adoption of the AJAX in sites such as Gmail, Google Maps, Amazon, Yahoo and Flickr.

The key to Ajax is the XMLHttpRequest object. This object is created as a result of a call to a JavaScript function after a user-generated event such as a mouse click. The object is configured with a request parameter that includes the ID of the component that generated the event and any value that the user might have entered. The XMLHttpRequest object sends a request to the web server, the server processes the request, exchanges any data required from the data store, and returns data such as a XML document or plain text. Finally the XMLHttpRequest object triggers a callback() function to receive and process the data and then updating the HTML DOM to display any outcome (Draganovam, 2007). Where Ajax is unique is that the request that is sent to the server asynchronously allowing the user to continue to interact with the user interface whilst the request is being processed.

This sequence of events differs from the classical model of the Web where a user action such as a mouse click triggers a HTTP request containing the URI of a desired resource (usually a web page) to a web server. The web server accepts the request, in some cases does some server side processing and data retrieval, and then returns a complete HTML page back to the client to be displayed through the browser.

The classical model of the Web implements a turn-taking protocol in which users must wait for the reply after they submit a request. The reply results in a full-page refresh of the browser in order to display the result of the request. The model is simple, well understood and effective. Nielsen notes "Users are in control of their own user experience and thus focus on your content" (Nielsen, 2007).

Ajax eliminates any delays caused by turn-taking by introducing an Ajax engine into the browser between the user and the Web server. Results of requests, e.g. XML data, can be loaded into the Ajax engine allowing the user to interact with the data without having to communicate with the server. The engine can make additional requests to the server for new data or interface code asynchronously without blocking the user from

continuing to interact with the web page. In particular, the JavaScript routine running in the browser can update only the necessary components of pages that require updating without a full-page refresh (Zucker, 2007; Kluge et al, 2007).

Ajax addresses the limitations in the classical model of the Web in two ways:

1. Enhancing response rates through data buffering. Ajax supports predictive downloading that allows data to be requested and buffered before the user needs it. Preloading of data is based on likely user actions in the context of the current page status. For example, Google Maps will automatically preload the regions of the map adjacent to the current views enabling the user to pan the map without any pause occurring as new sections are downloaded (Zucker, 2007). Another common use of data buffering is to support dynamic filtering allowing users to interact with multiple form options without the need for continuous page refreshes.

2. Enhanced user interactivity through asynchronous communication. The capacity of Ajax to update only relevant components of Web pages provides developers with the ability to create new interaction models. For example, Gmail uses Ajax to enable a new email message to be displayed in the interface when it is received without the need for the whole page to be updated. This feature enables Gmail to appear to the user to be acting more like a desktop application than a Web interface (Zucker, 2007). In addition, Ajax enables developers to present to users a range of innovative and engaging widgets and screen components that surpass the traditional controls available through HTML such as checkboxes, radio buttons, form fields and buttons.

Data buffering and asynchronous communications facilitate innovative Web applications that can be designed to be substantially more fluid than before resulting in less interruptions to the workflow of the user (Kluge et al, 2007). Oulasvirta and Salovaara (2004) suggest that interfaces should be 'invisible' without interruptions that cause switching of attention from a task as these can hamper memory and higher-level thought processes involving heavy load for working memory, for example when solving novel problems. A well designed Ajax-based Web application that avoids pauses, delays and interruptions may be able to provide the optimal experience of 'flow' that can result in engagement, enjoyment and increased productivity (Chen et al, 2000).

3 AJAX - Challenges

3.1 Usability Principles and Disorientation

Nielsen (2005) has suggested ten general heuristics for user interface design such as 'visibility of system status', 'match between the system and real world', 'consistency and standards' and 'recognition rather than recall'. Nielsen (1993) also recommends five usability attributes that include 'learnability', 'efficiency of use' and 'memorability'. Other research has produced similar sets of principles for the design of usable computer systems such as Dix et al. (1997) who suggested key attributes including 'learnability', 'flexibility' and 'robustness', and Shneiderman and Plaisant (2005) who proposed heuristics

including: ‘strive for consistency’, ‘offer informative feedback’ and ‘reduce short-term memory load’.

The general principles of consistency, learnability and feedback are common themes that are relevant when considering the usability of commercial Ajax-based Web applications.

Consistency: Cognitive psychologists suggest that as we interact with the world our mind constructs mental models of how things work (Johnson-Laird, 1983). Mental models may be used to anticipate events, to reason, and to explain the world. Interfaces that align with a user’s mental model will support their predictive and explanatory abilities for understanding how to interact with the system (Norman, 1988). Conflicts between the user’s mental model of a system and the reality of how a system behaves can result in disorientation and/or cognitive overhead. We would expect that the classical page-based model with the turn-taking protocol has become entrenched as part of a user’s mental model of the Web.

Learnability: A basic principle of Human Computer Interaction (HCI) is that user interfaces should be easy to use and predictable (Shneiderman and Plaisant, 2005). This is particularly important for commercial web-sites as we know that in general, Web users are impatient, require instant gratification and will leave a site if they cannot immediately figure out how to use it (Nielsen, 2000).

Feedback: Norman’s theory of affordance (1988) tells us that an interface should provide inherent clues to what actions are possible at any moment, the results of actions and the current state of the system so that users will know what to do instinctively. The success of the enhanced interactivity enabled through Ajax relies on the designer’s ability to provide appropriate feedback on the status of the Web application at all times.

A lack of consistency, learnability and feedback can result in disorientation and cognitive overhead in the users of Web applications. Conklin (1987) described disorientation as “the tendency to lose one’s sense of location and direction in a non-linear document” and cognitive overhead as “the additional effort and concentration necessary to maintain several tasks or trails at one time”. Disorientation and cognitive overhead are issues that have been thoroughly investigated in traditional hypertext systems.

3.2 AJAX Usability

Ajax can bring many benefits to the usability of the web applications by making the user interface more interactive and responsive. However, use of Ajax techniques has some challenges for achieving and/or maintaining usability. Nielsen (2007) notes that many Ajax-enabled Web 2.0 sites are “neglecting some of the principles of good design and usability established over the last decade”.

The page-based model of the Web is well entrenched as it provides the user’s view of the information on the screen, the unit of navigation (what you get when you click), and a discrete address for each view (the URL). The user’s mental model of how the Web operates has created a strong expectation that each interaction will result in a brief delay followed by a full refresh of the entire page. The simplicity of the original page-based

model of the Web contributes to its ease of use and its rapid uptake (Nielsen, 2005).

Ajax “shatters the metaphor of the web page” (Mesbah & van Deursen, 2009). With Ajax, the user’s view is determined by a sequence of navigation actions rather than a single navigation action (Nielsen, 2005). The asynchronous client-server communication in Ajax may result in surprises for users as updates to a web page may occur without any user interaction. Users may also be surprised as individual sections or components of web pages are updated without a full-page refresh or without any user interaction. New innovative controls and widgets might appear on web pages providing features or functionality not normally found on web sites and without any clues to their operation. Finally, the user may find that particular features within the browser might not respond as expected such as the back button, forward button, history list, bookmarks, saving, printing and search.

The focus of this investigation was on two particular usability issues relating to Ajax implementations: inconsistencies in the operation of the Back button and the management of updates to web pages.

3.2.1 Issue 1: Back Button

There has been a substantial amount of empirical research that has investigated the use of the browser’s ‘Back’ button and the page revisitation behaviour of users. For example, studies that used client-side logging of user actions when using the Web found that dominant navigation choices were embedded links (52%) followed by the Back button (41%) (Catledge & Pitkow, 1995), and that the Back button was used for approximately 40% of user actions (Tauscher & Greenberg, 1996).

The major paradigm challenge for Ajax technologies is the unpredictable behaviour of the Back button on the browser. Since an Ajax application resides in a single page, there is sometimes no page to return to, or no page history to navigate resulting in unexpected outcomes for users (Rosenberg, 2007). Nielsen (2005) in his article entitled ‘Ajax Sucks’ noted that “the Back feature is an absolutely essential safety net the gives uses the confidence to navigate freely in the knowledge that they can always get back to firm ground. Breaking it is a usability catastrophe”.

The lack of state information resulting from asynchronous data exchange in Ajax applications also affects the user’s ability to use the Forward button and history list with confidence. Similarly, the outcomes of bookmarking a page from an Ajax application can be inconsistent with users expecting a bookmark to return a particular page status however frequently the bookmark will only return the initial screen of the Ajax application (Kluge et al, 2007).

There are several technical solutions to overcoming the Back button issue. For example, Google Maps artificially inserts entries into the history list so that when the Back button is clicked, the application state is reverted to simulate the expected ‘back’ behaviour. However there appears to be no generally accepted solution to this issue.

3.2.2 Issue 2: Update Management

One of the most powerful features of Ajax is the ability for designers to implement functionality that causes a particular component or section of a web page to be updated rather than a full-page refresh. These 'part-page' updates can be implemented to occur either asynchronously without any user action such as receipt of an email in Gmail, or in response to a user interaction such as a mouse click. There are two related usability issues that can result from part-page updates.

The first issue is linked to the user's awareness that an update is occurring. Full-page refreshes in classical page-based interactions usually result in the browser displaying a visual indicator that informs the user that processing is occurring and that a new page is loading. For example, Internet Explorer has a solid circle that spins to indicate that processing is taking place, Firefox displays small spinning circles with different colours and Google Chrome has a half-circle that spins. Ajax applications cannot utilise the standard browser-based loading indicators. The default in Ajax is that no indicator is provided which can result in usability problems with Tonkin claiming "without explicit visual clues, users are unlikely to realize that the content of a page is being modified dynamically" (Tonkin, 2006). It therefore falls to developers to implement visual clues into their Ajax to inform the user that processing is occurring and also when the update has completed. Rosenberg (2007), reporting on the redesign of Yahoo Mail noted that there are no real standards for progress indicators in Ajax. Likewise, there is no standard approach to inform the user that the update has completed. Practices appear to range from sites that simply stop the loading indicator when the Ajax processing is completed whilst others display an 'Update Completed' or similar message. The potential for inconsistencies in how Ajax updates are reported to users could result in user disorientation.

The second issue relates to the user's awareness of the nature and/or location of the actual change that has occurred on the page after a part-page update. Nielsen (2007) notes "users often overlooked modest changes such as when they added something to a shopping cart and it updated only a small area in a corner of the screen". This effect is linked to a psychological phenomena called 'change blindness' where humans might not notice visual changes, even when these are "large, anticipated, and repeatedly made" (Resink, 2002). This effect has also been referred to as 'attentional gambling' where there is some uncertainty regarding where a user's attention should be focused (Hudson, 2007). Once again, the potential for users to overlook changes as a result of part-page updates could result in usability problems.

4 Experiment

The usability challenges of Ajax described in the previous section have been documented in the research literature with some developer forums containing various technical solutions that could be employed to alleviate any undesirable usability effects in their Ajax applications. What is unclear is how commercial Ajax developers have actually responded to these issues or if Ajax enabled web

sites continue to exhibit undesirable behaviours that might result in user disorientation and cognitive overhead.

An empirical investigation into the usability of commercial Ajax-based web applications was undertaken to examine the impact of these usability issues. The specific issues to be investigated included the consistency of the operation of the Back button and whether the management of part-page updates affected the user's experience.

5 Method

Twenty students and staff from Swinburne University of Technology (6 female and 14 male) participated. Their age groups varied from 18 to 50 years. Participants were recruited from all academic disciplines using notice board advertisements. Participants were tested individually in a specialist usability laboratory and were paid a small fee for their time. Ethics approval had been received prior to conducting the study.

A repeated-measures design was used in which participants each completed two tasks on each of four commercial web sites that employed Ajax-based web technologies. The sites selected for the study were four popular hotel booking sites that incorporated various aspects of Ajax including part-page refreshes and innovative user controls. The sites were coded as O for orbitz.com, T for tripadvisor.com, K for kayak.com and H and hotels.com. The participants were provided with written instructions describing the tasks to be performed with the order in which the sites were presented to the participants being counterbalanced. The tasks involved finding a list of suitable hotels that were available in a particular city, on a particular date, in a particular neighbourhood, with a ranking of 4 or 5 stars and containing a restaurant. The participants were instructed to find hotels in two different cities, Paris and then London. Participants were encouraged to speak aloud as they completed the tasks. The participant's actions and comments were captured using Morae Recorder and were analysed through Morae Manager to establish search times and other patterns of use. The participants completed a System Usability Scale (Brooke, 1996) after using each hotel booking site to assess views on learnability, design and overall satisfaction when using each site.

6 Results

6.1 Task Completion Times

Task completion times were operationalised as the time taken from when the initial list of hotels were displayed after the participant had selected the desired city and dates, until the resultant list of hotels were displayed. This approach measured the time taken for the participant to apply filters to the star rating, neighbourhood and restaurant settings. The outcome of changes to each filter control caused a part-page refresh of the hotel list.

Figure 1 presents the separate task completion times for each major city with the total time being the sum of the task completion times for both city tasks.

A visual examination of Figure 1 suggests that the overall task completion times on Site K and Site O were

shorter than the times for Site H and Site T. In addition, the results show that the completion times for the initial task (Paris) for each site was greater than the time to complete the task for the second task (London). This is expected, as we know that users generally perform better once they have gained an initial familiarity with a system. This is particularly evident in Site H.

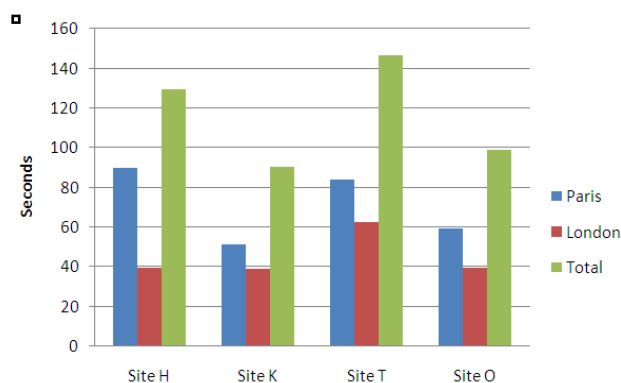


Figure 1: Task Completion Times

A set of one-way repeated measures ANOVAs were conducted to compare the time to complete the total booking time for both tasks for each test site as well as each booking task separately. There was a significant effect for total booking time, Wilks' Lambda = .29, $F(2, 19) = 13.78$, $p < .0005$. A pairwise comparison found that the total booking time for Site H was significantly different from Site K ($M = 39.00$, $p < .05$) and Site O ($M = 30.50$, $p < .05$). The analysis also found that Site T was significantly different from Site K ($M = 56.25.0$, $p < .05$) and Site O ($M = 47.75$, $p < .05$).

There was also a significant effect for the completion time of the initial Paris task, Wilks' Lambda = .37, $F(2, 19) = 9.87$, $p < .001$. A pairwise comparison found that the Paris task for Site H was significantly different from Site K ($M = 38.80$, $p < .05$) and Site O ($M = 30.80$, $p < .05$). The analysis also found that Site T was significantly different from Site K ($M = 32.85.0$, $p < .05$) and Site O ($M = 24.85$, $p < .05$).

This analysis suggests that Site K and Site O provide better support to users in completing timely bookings in comparison to Site H and Site T. This difference is particularly significant when considering only the initial task (Paris) suggesting that Site K and Site O provide a more supportive experience for users interacting with the site for the first time.

6.2 System Usability Scale

Figure 2 shows the overall results of the System Usability Scale (SUS) indicating a high level of overall satisfaction with Site K and a lower level of satisfaction for Site T. Wilks' Lambda = .09, $F(2, 9) = 24.87$, $p < .0005$. A pairwise comparison found that the SUS scores for Site K were significantly different from Site T ($M = 12.00$, $p < .05$).

These results are consistent with the analysis of the completion times that found that Site T provided the least amount of support for users when completing bookings. There was also a significant preference for Site K that yielded the shortest task completion times.

The SUS questions were categorized into three groups according to the focus of the question:

- Using: questions such as "I like to use this site" and "I felt confident using this site"
- Learning: questions such as "I needed prior learning" and "assistance was required"
- Design: questions such as "site was complex" and "too much inconsistency".

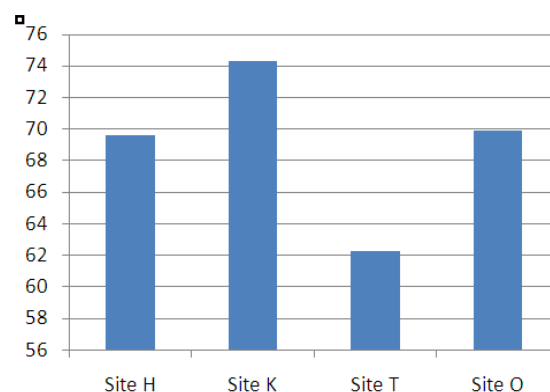


Figure 2: System Usability Scale Ratings

Figure 3 shows the results of the System Usability Scale based on the categorisation of the questions into 'Using', 'Learning' and 'Design'. The results suggest that Site T was regarded by participants as a site that required previous experience and assistance. It was also the site that was least preferred. Site K was identified as the site that was preferred in terms of overall design of simplicity and consistency. These results are consistent with the analysis of the task completion times.

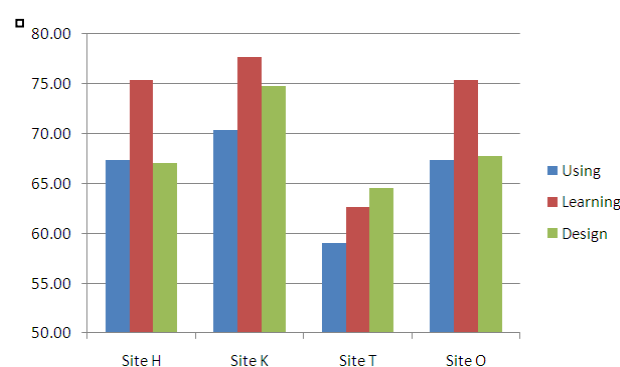


Figure 3: System Usability Scale Ratings by Category

6.3 Back Button Use

There were five steps required to complete each booking task. The steps required filters to be set for the selection of the city and dates, neighbourhood, amenities, rating and then choice of hotel. Participants generally used the browser's Back button to return to the site's home page after viewing the list of matching hotels. The Morae Recorder captured the participant's actions and their verbal comments when navigating back to the home page. Each web site provided a different user experience when the Back button was clicked.

Site H: The Back button performed predictably with each click stepping back through each previous status of

the Ajax application essentially undoing each search criteria in the reverse order that they were applied. To the user, this appeared to be stepping back one 'page' at a time and hence performed according to the classical model of the Web.

Site K: The Back button performed unpredictably. In most instances each Back click had no effect with the majority of the participants giving up and either clicking on the site's logo to return to the home page or re-entering the URL. P15 clicked Back 13 times finally giving up and stating "Basically the Back button in the browser does not work". The function of the Back button did not perform as expected and confused many participants.

Site T: The Back button was somewhat predictable. Each click appeared to go back one 'page' eventually returning to the home page, however there was no change to the filter settings after each click that was noted by several participants. 8 participants abandoned the use of the Back button after 3 or 4 clicks and clicking on the logo instead, possibly due to the lack obvious change in the page after each click. The level of feedback on each click was clearly not consistent with the classic web model.

Site O: The Back button performed relatively predictably. The majority of the participants found that when they clicked the Back button the search results page was displayed with all search criteria removed. When they click Back a second time the home page was displayed. For 7 participants there was no function for the Back button at all. Some of the participants noted that the first click removed all the search criteria, P6 stating "it should inform the user that when you click Back everything will be cleared", whilst P18 stated "If you went back to a previous page then you would have to remember all the criteria you put in or re-select all the criteria as it is all lost". Some participants expressed surprise that the second click of the Back button returned immediately to the home page.

6.4 Update Management

Each hotel booking site provided different approaches to: (i) indicating the request is being processed, and (ii) indicating the request is complete with a part-page refresh of the component of the page containing the list of matching hotels.

Site H: The relevant filter control is highlighted with a yellow background and is adjacent to a small spinning circle similar to the Internet Explorer processing indicator. After the part-page refresh the vertical scroll position of the page was reset to the top of the window clearly indicating that the refresh had concluded.

Site K: A pop-up box appears stating "Updating results - Filtering by..." noting the particular criteria, e.g. star ratings or amenities. The box disappears when the refresh of the list is complete with the page remaining at the same vertical scroll position.

Site T: A pop-up box appears stating "Updating your results...". The box disappears when the refresh of the list is complete with the page remaining at the same vertical scroll position.

Site O: A pop-up box appears stating "Loading your results". A rotating circle similar to the Firefox

processing indicator is displayed in the box. The section of the page that is being updated fades with a white transparent overlay. The box and fade disappear when the refresh of the list is complete.

An analysis of the Morae recordings was conducted focusing on the participant's timings and comments whilst filters were being processed. The analysis examined in particular the participant's reactions to the processing indicator for each site and their responses when processing completed. Two usability issues emerged.

The first issue related to the visibility of the processing indicator. Site H highlighted the relevant filter control with a yellow background with a small spinning circle. This method of indicating processing was visually less obvious in comparison to the methods used in other sites and as a result it became apparent that many participants did not notice the indicator. P18 stated "Doesn't really tell you that it has done the selection criteria" whilst P20 noted "This site appeared to be reloading however did not give a more specific indication". The lack of an obvious processing indicator may have resulted in either the participants continuing to apply filters without waiting for the processing to be completed or pausing to try to establish the outcome of the application of the filter. This effect may have contributed to the longer completion times and the lower SUS scores for Site H in comparison to Site T.

The approach utilized by Site O involving a pop-up box with a white translucent overlay of the results section of the page. This combination provided the most obvious processing indicator. It was apparent from the analysis that the majority of the participants paused until the refresh was completed before continuing any interaction. The relatively high SUS scores may suggest that this approach is effective in providing feedback to users. This approach may have resulted in slightly longer booking times than Site K due to the enforced pause with P11 stating "At least you know it's filtering but it's slower".

The second usability issue related to the participant's awareness of the status of the system when processing had completed. A significant issue that was observed in Site T related to a particular filter control that could only be viewed on a standard window size by scrolling down. It was observed that many participants scrolled the page to the top of the screen immediately after interacting with this particular filter and therefore missed seeing the processing indicator pop-up box. As a result many participants appeared unaware that processing was occurring, e.g. P6 stated "Is it automatically filtering - I cannot see any change here" and P10 stated "I cannot tell whether it has done the filtering or not". This effect may have contributed to the longer completion times and the lower SUS scores for Site T.

A similar issue arose for Site K in cases where the popup box appeared only very briefly due to short processing times. For example, P13 stated "A little box came up really fast and I suppose this was telling me that it was changing the results". P11 wrote in his SUS feedback form that he did not believe that the site provided appropriate information to inform that the search had completed: "the filtering sign popped up over the screen and then it was very quick". P13 said "A

message flashed up but I didn't always see it" and P17 stated "There was no clear indicator that the new search results had changed. You had to be very aware of the page to notice". Whilst many participants expressed these concerns it is noted that Site K provide the highest SUS score and lowest task completion times.

6.5 Other Results

The following comments were noted regarding the design of Site K and Site O that had the lowest task completion times. P19 stated regarding Site K: "I like this, neat and clean looking – my favourite". In relation to Site O, P4 stated that "I think this site is cool" and P11 stated "The best site, slower when loading filtered data but clear and easy to use". Participant's comments suggested a strong preference for the design of Site K consistent with the categorized SUS scores.

There were multiple comments regarding the 'Rating' filter control implemented on Site H. The control was an Ajax-enabled 'slider' that required the user to move markers along a vertical line to select their minimum and maximum ratings. Comments included: P17: "Star rating is little bit annoying as you have to know how to use these bars", P5: "Change the rating control to use a select box", P7: "The Star bar is awkward and cumbersome to use", P8: "The star checkbox need to be improved", and P19: "The star rating feature was very awkward".

Many participants noted difficulties with navigating Site T, particularly locating several of the filter controls that were placed towards the top-right of the page template. Comments included: P6: "Move the location control to the left side", P9: "The neighbourhood location was difficult to find", P11: "the filtering options on the top of the site were confusing and awkward to find out first", and P17: "It was a little hard to navigate, was a bit annoying and it could make some of the options a bit easier to find". This design issue may have contributed to the longer completion times and the lower SUS scores for Site T.

7 Discussion

This study has examined three aspects of a set of Ajax enabled commercial websites to determine whether the known usability issues are apparent and if they have an effect on the usability of the sites. Table 1 shows a 'Traffic Light' summary of the results of the study. The two issues that were investigated are presented on the left side of the horizontal axis, i.e. the action of the back button, the effectiveness of the processing indicator along with ratings based on the general design of the site. On the right are the two performance indicators, i.e. the task completion times and the system usability scores (SUS).

The summary results suggest that Site K performed the 'best' with the shortest completion times and highest SUS ratings however when using this site the Back button performed unpredictably. There were also issues relating to the processing indicator where some participants were unaware that the Ajax had finished processing. These issues may have been compensated by the design of the site including good navigational support with participants expressing a clear preference for the site design in verbal

comments and the design related questions in the SUS questionnaire.

Site	Back Button Action	Processing Indicator	Design	Completion Time	System Usability Score
H	Predictable	Not obvious	Filter Control	129	70
K	Unpredictable	Brief	"Clean"	90	74
T	Somewhat predictable	Brief	Layout	146	62
O	Predictable	Very obvious	"Cool"	99	70

Table 1: Traffic Light Summary of Results

Site O was ranked the next 'best' having a predictable Back button, the most obvious processing indicator and a preference for the design. The slightly longer completion times may have been a result of the enforced pauses as a result of the processing indicator.

Site H had the least obvious processing indicator and issues with the Ajax-enabled 'slider' control for the star rating. Together these may have contributed to the longer completion times.

Site T had issues with both the predictability of the Back button and the timing of the processing indicator with many participants who missed seeing the indicator commence or finish. Whilst these may have contributed to the long completion times and low SUS ratings, the general site layout caused some frustration with many participants having difficulty in locating several filter controls which were placed towards the top-right of the page template.

The results of the study confirm that commercial web developers are inconsistently managing known usability issues. The results also indicate that combinations of the usability issues do affect user performance and satisfaction.

Section 3 of this paper describes the general usability heuristics of consistency, learnability and feedback. These general principles are common themes that are relevant when considering the usability of commercial Ajax-based Web applications. The results of this study may be considered in relation to these themes in order to generate some recommendations for web site developers who employ Ajax technologies.

Consistency: The operation of the Back button has now become entrenched as part of the mental model of Web users. This model allows users to interact with the Web with minimal cognitive overhead as they can confidently predict the outcomes of their actions and plan and execute browsing strategies. The operation of the Back button in two out of the four sites in this study broke the classical model of the Web resulting in some participants reporting confusion and disorientation. The results of the study may indicate that good navigational support and site design can alleviate the detrimental effects of an unpredictable Back button, i.e. users may not need to backtrack as much whilst navigating. The results suggest retaining Back button functionality consistent with the classical Web model is an important usability factor in conjunction with other factors. It is recommended that Ajax developers implement technical

solutions that ensure that the Back button has predictable outcomes.

Learnability: The ability for users to quickly figure out how to use a web site is a critical success factor in user acceptance [18]. The study found several design issues in Sites H and T that resulted in longer completion times and low SUS scores, particularly for the initial task when users were first exposed to the site. The effect in Site H was particularly profound as a result of an innovative Ajax 'slider' control (Figure 4).

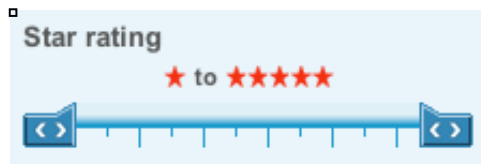


Figure 4: Slider Control in Site H

Bosworth (2005) captures this issue by stating “click on this non obvious thing to drive this other non obvious result”. Frequent users may learn how to use innovative Ajax controls with a possible improvement in performance but the negative effect on first time or novice users could outweigh perceived benefits. “Users need to be able to anticipate what a widget will do, appreciate how to interact with it, monitor changes during use and evaluate what has changed after use” (Atyeo, 2006).

Feedback: Usable systems should provide inherent clues to what actions are possible at any moment, the results of actions and the current state of the system so that users will know what to do instinctively (Norman, 1988). The four sites examined in the study implemented different approaches to indicating that a request was being processed and when the request was complete. The challenge of indicating to the user that there has been a part-page refresh of a particular component of the page was not managed well by two of the sites resulting in some user confusion and a decline in performance. Users appeared to favour Site O that provided very clear processing indicators. Ajax developers should employ standard processing indicator devices to clearly inform the user of the processing status. In addition Ajax developers should be aware of the potential for ‘change blindness’ that may be caused when a user is not aware of any change during a part-page refresh. The success of the enhanced interactivity enabled through Ajax relies on the designer’s ability to provide appropriate feedback on the status of the Web application at all times.

8 Conclusion

Ajax has several usability issues including consistency of the Back button and the management of part-page updates. These issues have been reported in the literature along with guidelines and technical solutions that that could be employed by Ajax developers to reduce undesirable usability effects in their Web applications.

This paper presents the results of an empirical study into four hotel booking sites that employ Ajax technologies in order to investigate how these sites have responded to the known usability issues. The results of the study were contrasted in relation to the general

usability principles of consistency, learnability and feedback.

The study found inconsistencies in how the sites managed the known usability issues and how combinations of the issues have a detrimental effect on user performance and satisfaction. The paper makes several recommendations to Ajax developers to in relation to consistency, learnability and feedback.

9 References

- Ankolekar, A., Krotzsch, M., Tran, T. and Vrandecic, D. (2007): The two cultures: Mashing up Web 2.0 and the Semantic Web, *Proc. International Conference on the WWW*, 825-834.
- Atyeo, M. (2006): Ajax and Usability, CapCHI, October 2006, Ottawa, From: www.capchi.org/pastactivities0607.html#oct06.
- Berners-Lee, T. (1989): Information Management: A Proposal, World Wide Web Consortium, From: www.w3.org/History/1989/proposal.html.
- Bosworth, A. (2005): Alex Bosworth’s Weblog: Ajax Mistakes, May 2005. From: <https://alexbosworth.backpackit.com/pub/67688>
- Brooke, J. (1996): SUS: a "quick and dirty" usability scale. In P. Jordan, B. Thomas, B. Weerdmeester, & A. McClelland. *Usability Evaluation in Industry*. London: Taylor and Francis.
- Catledge, L. D. and Pitkow J. E. (1995): Characterising Browsing Strategies in the World Wide Web, *Computer Networks and ISDN Systems*, 27:1065-1073.
- Chen, H., Wigand, R. T., and Nilan, M. (2000): Exploring Web users’ optimal flow experiences, *Information Technology & People*, 13(4):263 – 281.
- Conklin, J. (1987): Hypertext: An Introduction and Survey, *IEEE Computer*, 20, 9, 17-40.
- Dix, A., Cowen L. (2007): ‘HCI 2.0? Usability meets Web 2.0’. Panel position paper. *Proc. HCI2007*, British Computer Society.
- Dix, A., Finlay, J., Abowd, G., and Beale, R. (1997): *Human-Computer Interaction*, 2nd ed., Prentice Hall, Glasgow.
- Draganovam C. (2007): Asynchronous JavaScript Technology and XML, *Proc. 11th Annual Conference of Java in the Computing Curriculum*, London.
- Garrett, J. J. (2005): Ajax: A New Approach to Web Applications, Essay: Adaptive Path, From: www.adaptivepath.com/publications/essays/archives/000385.php.
- Hudson, W. (2007): Ajax Design and Usability, *Proc. 21st BCS HCI Group Conference*, HCI 2007, 3-7 September, Lancaster University, UK
- Johnson-Laird, P.N. (1983): *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge University Press.
- Kluge, J., Kargl, F. and Weber, M. (2007): The Effects of the AJAX Technology on Web Application Usability, *Proc. 3rd International Conference on Web Information Systems and Technologies*, Barcelona, Spain.

- Mesbah, A. and van Deursen, A. (2009): Invariant-based automatic testing of AJAX user interfaces, *Proc. IEEE 31st International Conference on Software Engineering*, 210-220.
- Millard, D. and Ross, M. (2006): Web 2.0: Hypertext by any other name?, *Proc. Seventeenth Conference on Hypertext and Hypermedia*, Denmark, 27-30.
- Nielsen, J. (2005): 10 Usability Heuristics, Use-it Alertbox, From: www.useit.com/papers/heuristic/heuristic_list.html
- Nielsen, J. (1993): *Usability Engineering*, AP, N.Y.
- Nielsen, J. (2000): *Designing Web Usability: The Practice of Simplicity*, New Riders Pub., IN.
- Nielsen, J. (2007): Web 2.0 'Neglecting good design. BBC News. 14 May 2007, From: news.bbc.co.uk/1/hi/technology/6653119.stm
- Norman, D.A. (1988): *The Design of Everyday Things*, Doubleday, N.Y.
- Oulasvirta, A. and Salovaara, A. (2004): A Cognitive Meta-Analysis of Design Approaches to Interruptions in Intelligent Environments, *Proc. SIGCHI Conference on Human Factors in Computing Systems*, 1155-1158.
- Rensink, R. A. (2002): Internal vs. External Information in Visual Perception, *Proc. ACM Symposium on Smart Graphics*, Hawthorne. N.Y. 63-70.
- Rosenberg, G. (2007): A Look into the Interaction Design of the New Yahoo! Mail... and the Pros and Cons of AJAX, *ACM Interactions*, Sept-Oct, 33-34.
- Shneiderman, B. and Plaisant, C. (2005): *Designing the User Interface - Strategies for Effective Human-Computer Interaction*, 4th Ed. Addison-Wesley, Reading, MA.
- Tauscher, L. and Greenberg, S. (1996): Design Guidelines for Effective WWW History Mechanisms, *Proc. Second Conference on Human Factors and the Web*, Microsoft Corp., Redmond, WA.
- Tonkin, E., (2006): Usability Issues for Ajax, Uklon Briefing Paper 94, From: www.ukoln.ac.uk/qa-focus/documents/briefings/briefing-94/html/
- Zucker, D. F. (2007): What Does AJAX Mean for You? *ACM Interactions*, Sep-Oct, 10-12.