RESEARCH ARTICLE

Using qualitative eye-tracking data to inform audio presentation of dynamic Web content

Andy Brown, Caroline Jay and Simon Harper
School of Computer Science
University of Manchester
Kilburn Building, Oxford Road
Manchester, UK
andrew.brown-3 — caroline.jay — simon.harper@manchester.ac.uk
(Received 00 Month 200x; final version received 00 Month 200x)

Presenting Web content through screen readers can be a challenging task, but this is the only means of access for many blind and visually impaired users. The difficulties are more acute when the information forms part of an interactive process, such as the increasingly common ‘Web 2.0 applications’. If the process is to be completed correctly and efficiently it is vital that appropriate information is given to the user at an appropriate time. Designing a non-visual interface that achieves these aims is a non-trivial task, for which several approaches are possible. The one taken here is to use eye-tracking to understand how sighted users interact with the content, and to gain insight into how they benefit from the information, then apply this understanding to design a non-visual user interface. This paper describes how this technique was applied to develop audio interfaces for two common types of interaction — auto-suggest lists and pop-up calendars. Although the resulting interfaces were quite different, one largely mirroring the visual representation and the other not, evaluations showed that the approach was effective, with both audio implementations effective and popular with participants.

Keywords: Web 2.0; AJAX; Visual Disability; Eye-Tracking

1. Introduction

All members of society need to interact with, and be kept informed of events within, their social network and their local and wider communities. Increasingly this is happening through digital media, particularly the Web, and if people are to avoid becoming ‘digitally excluded’, these media must be accessible to them. Yet this is a significant challenge, with exclusion resulting from any of a variety of causes. One of these is disability, and an important group of disabled people are those with visual impairments. The World Wide Web might be increasingly multimedia, but is still predominantly a visual medium. Enabling access to the Web through sound is an important task if people who are blind or visually impaired are to be ‘digitally included’.

While it may be argued that Web accessibility should be improving with the increasing sophistication of assistive technologies, and education and outreach efforts from the accessibility communities, it is not clear that this is, in fact, happening.
Indeed, it is perhaps the case that the proliferation of user-generated content and developments in mainstream Web technology mean that accessibility is lagging behind. It is the latter that is perhaps the most concerning: developers are using the simple building blocks of HTML and JavaScript to produce sophisticated pages with complex user-interfaces, such as mail clients, spreadsheets and word processing, that resemble their desktop equivalents. As these websites become ever more sophisticated and interactive, the danger is that these sites become even harder for those with visual impairments to use.

One of the key attributes that allows pages to behave like stand-alone applications is the ability to update part of the page without reloading or re-rendering the remainder. Thus, a user might click on a link (which might have been styled to resemble a button) and instead of loading a new page, one section of the current page changes. Several technologies may be used to achieve this; one common technique is AJAX (Asynchronous JavaScript and XML), which allows behind the scenes communication between page and server. While fully-fledged Web applications are in the vanguard when it comes to using these technologies, they are actually widely used in those smaller interactive sections of pages known as ‘widgets’. Two very common widgets are the auto-suggest list (ASL) and the pop-up calendar (or ‘date picker’).

An ASL is a list that provides the user with suggested entries for a text field, based on what they are typing. The suggestions appear when the user starts typing in the box (either after the first character or after a set minimum, e.g., 3), and are refreshed for each change. They are used in both free-form input fields (e.g., for search queries) and for those where entry is constrained (e.g., a list of railway stations). Figure 1 gives an example of the latter. ASLs are both common and of a relatively consistent form. A brief survey looked at the Alexa global top sites. After removing duplicates (e.g., www.google.com and www.google.de) the remaining top 20 were examined and found to contain 12 instances of ASLs.

Pop-up calendars are common on forms requiring date input. They typically display one month in a tabular format; Figure 2 gives some examples. The month is given at the top, with controls on either side to navigate to the next or previous month; the table below has a header row giving the days of the week, followed by the dates represented by numbers with one week per row. The table cell representing the date currently in the input box is often given visual highlighting. A typical mode of operation (used by all the examples in Figure 2) is to have a text input

---

box that, when focused, initiates the appearance of a calendar. The calendar usually appears immediately below the text box, and over other content (hence ‘pop-up’). The user then has the choice of typing the date manually, or selecting the date using the calendar. The latter is usually done with the mouse, although some are keyboard accessible. Once a date has been selected, the calendar disappears.

There are two main reasons for these type of widgets being inaccessible. First, many screen readers are poor at either identifying page updates or notifying the user when they occur (particularly if the screen reader is old or the page does not use WAI-ARIA markup). This means that it appears to the user as if nothing has happened. The second problem is the fact that widgets such as calendars and ASLs are created using standard HTML components, then given style and functionality through scripting. The fact that these components are handled by assistive technologies means that the widgets themselves may be considered accessible. This is superficial accessibility, however, as the widget is often not usable in an efficient way via assistive technologies. The reason for this is that sighted users are able to perceive the layout, components and styling at once, and this combination often makes the purpose and use of a widget explicit. For those people who don’t have access to the layout and styling, however, this information is implicit, and can be very difficult to infer. In this case, users do not have the advantage of prior understanding when starting an interaction; unlike sighted users it can be difficult to know what to expect, so manually identifying the results of an action can be hard.

How then should one approach making these widgets accessible? Is it really necessary — after all it is possible to enter search queries, dates and so on manually? We contend that it is indeed necessary, and that a successful approach is to examine how sighted users — those for whom the widgets were designed — use them: how do they interact with the widgets, and what benefits do they derive from using them? Only once these questions are answered is it possible to design an audio implementation of the widget for screen reader users. This paper describes how this
approach was successfully applied to both pop-up calendars and auto-suggest lists: eye-tracking studies were used to develop audio interfaces that were then evaluated by screen reader users. In section 2 we examine what approaches it is possible to take: what principles have other researchers used to create audio interfaces; we also introduce our approach. Section 3 describes the eye-tracking studies performed on sighted users and presents their results. This is followed in section 4 with a qualitative analysis of the results and a description of the audio implementations that these findings informed. Finally, sections 5 and 6 describe how these implementations were evaluated, using blind and visually impaired participants (and found to be popular and effective), and discuss the benefits and drawbacks of the approach.

2. Background

Dynamic updates are problematic for screen reader users. A review of how users of different assistive technologies deal with dynamically updating pages (Brown and Jay 2008) showed that older browser / screen reader combinations apparently did not even respond to some updates, in particular those which occurred automatically. Even newer technologies often did not notify the user that the page had changed (it should be noted that technology is changing rapidly, even if users do not always have the most up-to-date version). Given that updates take many forms, the problems causing this situation are two-fold: it is difficult both to detect sufficient information about the update (is it a newly inserted list, a list of suggestions, part of an advertisement, or some different information?) and to design an effective way to present an update.

There are two possible ways of understanding the nature of the content: to make it explicit, via markup, or to make inferences from the HTML. The former is likely to be more robust, as markup will be generated by the developers, who should have a good understanding of what the content is, but requires standardisation, and widespread implementation. This approach is being pursued by the W3C’s Web Accessibility Initiative, in the form of the Accessible Rich Internet Applications Suite (ARIA)1. The essence of WAI-ARIA (Gibson 2007) is that adding semantic metadata will allow the user-agent/assistive technology partnership to render the rich information appropriately. Although there are situations where it is difficult to find an appropriate solution through ARIA (see, for example, Thiessen and Chen 2007, Brown et al. 2009), there are also many situations where it can be applied effectively. Currently, however, ARIA use is limited, and despite its inclusion in some JavaScript libraries, there are still many sites whose dynamic content does not have ARIA markup. In these instances, the only option for assistive technologies is to make inferences about the interaction.

In either case, there remains the difficulty of designing an interface that allows the ‘widget’ to be used effectively and efficiently. It is on this aspect of the problem that this paper focuses. Perhaps due to its relative novelty, there hasn’t been much previous research into non-visual presentation of dynamic web content, but the different methods that may be applied to this ‘modal transformation’ may be gleaned by an examination of how researchers in other areas of accessibility have approached the problem. Researchers and developers have considered the design of audio interaction with similar content in other contexts — e.g., tables (Yesilada et al. 2004), lists (Pitt and Edwards 2003), Cartesian graphs (Mansur et al. 1985), and node-arc graphs (Brown et al. 2004) have all been tackled — and their solutions

1Introduced at http://www.w3.org/WAI/intro/aria.php
can form the basis for designing interaction techniques that can be applied to dynamic Web content.

A common approach in the field of accessibility is to make the information available through audio by direct translation. Following this technique would normally cause ASLs to be presented as a list. Typically, this would mean the list itself is announced (e.g., ‘unordered list with ten items’), followed by the items (suggestions, again with each suggestion announced as an ‘item’). In other cases (such as Google suggest), the suggestions are presented in a table — the left column gives the suggestion, while the right gives the number of ‘hits’ resulting from that search query. Calendars are generally presented as tables, which are announced in a similar way to lists (e.g., ‘table with seven columns and seven rows’).

The other common approach is to consider the benefits a form of presentation offers, and to enable those benefits in a non-visual form. This can be seen as an extension of direct translation. Instead of simply allowing the user to move element by element through the document, extra functionality is added that replicates some of the benefits. For example, tables are two-dimensional, and a key function is the ability to relate a value to its row and column headers (Yesilada et al. 2004, Oogane and Asakawa 1998). Similarly, the ability to overview information is seen as a crucial benefit for sighted users, and can be provided in audio for information such as tables (Kildal and Brewster 2006, Brown et al. 2003, Kildal and Brewster 2005) and numerical graphs (Brown et al. 2003).

We contend that simply translating the content is not always appropriate, even when done with an understanding of the benefits that style of content offers, indeed that it can often be a poor choice. The case of pop-up calendars is a good example: direct translation would give the updated content to the user to browse, and present this as a table. The user could move between the cells of the table until the desired date is found. As noted above, tables are difficult structures to navigate non-visually, but even with the enhanced functionality such as header information and overviews, it is still not clear that it would be an effective method for helping users select a date. Widgets, such as ASLs and pop-up calendars, are used interactively, and are likely, therefore, to require a rapid, more direct, form of interaction. What is needed, therefore, is not just to make the content accessible, but to make the process by which users understand and use the content accessible.

Our approach is essentially to ignore the underlying HTML structure of the content, but instead to attempt to understand how they are used by sighted users, and what benefits they bring. This can only be done in the context of the interaction process, such as completing an input box, or selecting a date. Once the benefits are understood, an audio implementation can be designed to offer those same benefits to screen reader users. In this approach, the visual presentation of the content, and particularly the underlying HTML structure, are of secondary importance; what matters is how the information is used.

Gaining the understanding of how sighted users view, use, and benefit from these dynamic regions of content is not a trivial task. As we have noted, theoretical considerations can be useful, but these are best combined with empirical research. One tool that can provide valuable information about what users view is eye-tracking. Eye tracking has been used to investigate cognitive processes for over 100 years (Rayner 1998), and is a powerful tool when used in conjunction with Web pages, enabling us to determine those areas that are most salient (attract the most fixations), and those that receive little attention. Eye tracking’s most obvious application is in improving the standard design and layout of Web pages, and evaluating their usability (Russell 2005). Studies have also examined the saliency of items on a page under varying conditions (e.g., Granka et al. 2004, Pan et al.
2004, Schiessl et al. 2003), how eye movements vary according to information scent (Pirolli et al. 2001) and how looking for a menu is influenced by page complexity and prior expectations (McCarthy et al. 2003). Eye-tracking is thus an excellent tool to help develop an understanding of the benefits of certain forms of presentation.

3. Eye-tracking Experiments

The purpose of the eye-tracking experiments was to gain an understanding of how sighted users interact with the dynamic content, to understand how the widgets were used, and how they benefited the users. The study tracked users’ eye-movements as they both browsed and performed a series of tasks, each of which was designed to prompt interaction with an element containing dynamic content.

3.1 Method

17 male and 13 female volunteers, aged between 18 and 34, took part in the study, which took approximately 30 minutes. All participants used the Internet on a daily basis. Participants sat in front of a 17” monitor with a built in Tobii 1750 eye tracker, and Tobii Studio Professional edition eye gaze analysis software was used to record and analyse the data. The data collected was limited to fixation data from the eye tracker (participants were not asked to describe or justify their actions). Analysis was performed retrospectively, to identify patterns of behaviour and formulate hypotheses about how people interact with different types of dynamic content.

During the experiment participants encountered up to 3 different instances of auto-suggest lists, as follows: the Google Suggest page with input boxes for general Web search, where matching suggestions give commonly used search terms; the Yahoo! home page with an ASL matching locations for weather forecasts, and; the Kayak travel site flight search engine, where an input box was used to enter a destination and suggestions contained a list of matching airports (see Figure 1). One example of a Calendar was encountered; this was on the Kayak site and was used twice, to select the departure and return dates (see Figure 4).

The task performed on the Kayak site was: ‘Search for a flight for one person from Manchester to New York, leaving on <date> and returning a week later’; as well as interacting with a suggestions list, this task required the user to use two popup calendars, to find both a particular date and a relative date, thereby covering the majority of uses of a calendar. Google Suggest was used both for a directed task (‘Search for the University of Manchester’) and as a start point for undirected browsing; the Yahoo! home page was used as a start point for the task ‘What will the weather be like in New York today?', which sometimes resulted in participants encountering an ASL of Weather locations.

In all cases, interaction was documented with live Web sites, as this was necessary for many of the dynamic features to function, meaning that all participants viewed slightly different information\(^1\).

\(^1\)Note that this also means that the pages are no longer the same structure, and that the nature of some of the dynamic updates has changed or may change.
3.2 Results

The 30 participants encountered ASLs a total of 90 times during the experiments. Pop-up calendars were encountered by all but 4 participants, for whom the dates were correctly set when they loaded the page. The details of the interactions with each type of content are given below.

3.2.1 Auto-Suggest Lists

There was considerable consistency in the manner in which people used the ASLs. On the Kayak site, all 24 participants for whom suggestions appeared selected their destination airport from the list. The majority of participants viewed the Google Suggest ASL when it appeared: 9 selected from it in both tasks; 10 in one task. 6 participants did not select from the list, but did view it extensively for at least one task. Observation of gaze behaviour strongly suggests that the presence of the ASL, whether people select from it or not, is used to correct spellings, or reassure the user that they are typing a reasonable query. Only 3 participants appeared not to view the ASL at all. Of the 4 participants who encountered the ASL on the Yahoo! page, all viewed the list briefly, one of whom selected from it.

The results also showed that the first three suggestions received the majority of attention, with those below rarely being fixated. This pattern was consistent across the different ASLs, and was backed up by quantitative data (Brown et al. 2009). Figure 3 gives a flavour of the quantitative data, showing the fixation count and duration for the search box and the first three suggestions.

3.2.2 Calendars

Of the 26 participants who needed to input dates, 2 did not fixate the calendar that appeared and 1 viewed it then chose to edit the dates manually. Of the remaining 23, 14 requested the calendar directly by clicking the symbol to the right of the date field. 9 participants clicked in the date field initially to edit the departure date, but used the Calendar when it appeared, and of these, 6 then went on to click the symbol, to request the Calendar, for selecting the return date.

The eye-tracking not only gave information about whether participants looked at the calendar, but also about the way in which they used it. When selecting the departure date, they quickly scanned and located the required day. When selecting the return date, specified to them as ‘a week later’, people used the layout of the calendar to work it out. Participants treated the calendar as a table, running their mouse and/or gaze along the row underneath the departure date (which was
highlighted in red), before selecting the date directly underneath, or focusing on the departure date, and then moving down the same column by one row (see Figure 4).

4. Visual to Audio Translation

While the eye-tracking studies give information about what the users fixated, this information needs deeper analysis in order to understand how they benefited from it. Only once this has been done can audio implementations be designed (section 4.2) and evaluated (section 5).

4.1 Analysis

The eye-tracking experiments allow us to understand how sighted people interact with both types of dynamic content. First, it is notable that nearly all users fixated the content, and most used it. This despite the fact that it was not necessary in order for them to complete their tasks — clearly the content was seen as having some benefit. There were differences, however: while the majority of users decided to use the calendar to make the date selection, many ASL users only glanced at the suggestions then continued to type. The latter behaviour was thought to be for reassurance, confirming that the entry was a reasonable one, and was spelt correctly. The user benefits by being able to scan over a changing list of items, that give confirmation that their typing is reasonable, and to select an item in order to save typing. The fact that many users glance at the list, but do not select from it, suggests that receiving the information is more important than being able to interact with it. Importantly, the list does not interfere with the users primary task of typing. It is notable that selecting from the list appeared to be a more common behaviour when input was constrained (selecting an airport in the Kayak task) than when it was not.

The calendar provides an interesting contrast. In this case, the interaction is important, while much of the content is not. Users wish to locate the required date as rapidly as possible, then select it to complete the input field; the other content is not interesting in itself, but only serves to aid location by framing the desired date in a familiar layout. The tabular layout is beneficial because it is familiar, but more importantly because it allows relative dates to be determined quickly and more easily. Figure 4 shows one participant adding a week to a date by simply moving down a column. The interaction is important in this case, because it ensures the correct format without typing errors: the user does not need to consider
whether to use dd-mm-yyyy or mm-dd-yyyy or any of the other possible variations — one click and the date is set correctly.

Does direct translation provide a means for effectively presenting these updates non-visually? The actual structure of the ASLs varies according to the site. Although most are lists, the suggestions from Google are arranged into a two-column table, with the right-hand column giving the approximate number of search results each suggestion will return. Visually, however, this appears more as an annotated list. Direct translation is perhaps appropriate in this case, although refinements are required if the benefits of the visual form are to be retained. Crucially, the presentation needs to be brief and minimally intrusive, suggesting that the usual introductions (i.e., stating explicitly that this is a list) are best omitted. Furthermore, it is noted that presenting suggestions beyond the third is unnecessary.

The tabular structure of the calendar is a relatively complex one, and being two-dimensional can be difficult to read non-visually. Considering that the most important benefits of the pop-up calendar to sighted users are that it reduces errors by simplifying formatting and that it makes relative data calculations simple, it is not clear that direct translation of a table will make the most effective form for audio presentation and interaction. Wright (1981) noted that tables are cleverly convoluted lists, and this is particularly clear in the case of calendar tables. This suggests that, while a 2D tabular arrangement might suit sighted users, something closer to the 1D list arrangement might suit screen reader users. In this case, the challenges for the non-visual implementer are to handle formatting, and to facilitate movement over the list in such a way that date calculations are simplified.

This analysis of the eye-tracking data has given us the benefits that these types of content offer sighted users, and thus define what is required for an audio implementation. For auto-suggest lists, we must:

- Provide the top 3 suggestions.
- Not interfere with user input.
- Enable selection from the list.

For calendars, the requirements are:

- Automate formatting.
- Minimise potential for typing errors.
- Make relationships more explicit.

Once these requirements have been determined, the implementations can be designed. These do not need to relate to the visual representations, although they are necessarily limited by the information available within the update content.

4.2 Implementations

Analysis of the eye-tracking data strongly suggests that auto-suggest lists and pop-up calendars involve different forms of interaction, and that each requires a different approach when translating from visual to audio. While a relatively direct translation of the suggestions list should bring the benefits of an ASL to screen reader users, the calendars require something quite different from the tabular presentation used for sighted users. The user-interfaces of each are described below; details are limited to how the users interact with the update, not how the updates are detected, parsed, etc (Brown et al. 2010, gives some details). Implementation was in JavaScript, as an adaptation of the Fire Vox1 extension for the Firefox Web

1http://firevox.clcworld.net/
In order to understand the design of the user interfaces for the calendar and ASL, it is necessary to appreciate that they were designed as part of a wider system for audio Web browsing. This system was built specifically for evaluation and, as such, had a user-interface intended to be easy to learn and apply. Navigation was achieved using the 3 by 3 grid of number pad keys: the 3 rows allowed navigation at different levels (word, sentence and element) while the 3 columns allowed navigation backwards and forwards, or to speak the current location.

The visual form of the ASL appears, or changes, on each keystroke, and users glance at it either when typing, or during a pause in their typing. The audio implementation, therefore, involves rapid presentation of the list whenever the user pauses typing. When the user pauses typing, and the list updates (or appears for the first time), the suggestions are parsed from the HTML, and the first three are spoken. Speech is halted if the user continues to type (typing is usually ‘echoed’). The interface also allows users to browse the entire list: number pad keys 3 and 6 move to the next item, 1 and 4 to the previous item, and 2 and 5 speak the current item; the 7, 8 and 9 keys allow navigation to other elements as normal. The list is navigated in a loop, including all suggestions and the letters currently typed. If the user presses the enter key, the currently selected suggestion will be placed in the input field (accompanied by the message “Input set to [suggestion]”), and all number pad keys can be used to navigate as normal. The escape key also clears the list and allows standard navigation.

The audio implementation of the pop-up calendar also takes the form of a list, but is much more prominent and interactive than the ASL. In this case, the list is not presented as some background information that can be interacted with if desired, but becomes the most prominent part of the page — the focus. When the users navigates to a date field, and a calendar appears, the user is informed that this is a date entry field, and given the date that is already present in the field (or today’s date, if there is none). The user can then navigate through the calendar as a list to reach the desired date, at which point this is selected, and the date is entered into the input field using the correct format. This brings the benefits of automatic formatting that these calendars bring to sighted users, but does not help with relative date calculation. This is achieved by enabling navigation at three levels: day, week and month. Nine keys enable the user to move backwards or forwards a day at a time, a week at a time, or a month at a time, or to query the ‘current’ day, week or month (see Figure 5). Calendar interaction was performed modally — once the date entry field has been announced, the number pad keys used for general navigation are used to navigate the calendar. The mode is left automatically when a date is selected (by pressing the enter key), or manually with the escape button; a command is also available to enter this mode.

It can be seen that these will be quite different experiences for the user. Visually there are considerable similarities: the update is triggered by the user performing ‘normal’ activity (i.e., something that wouldn’t traditionally be expected to change the page), and appears in the form of an area of new content next to the input field and over the original surrounding content. Deeper investigation of the eye-tracking data, however, shows that they are not used in the same way. The audio interfaces that we have derived from this analysis differ in the following ways:

- The ASL is presented in the background, as extra information that appears when the user pauses typing; the calendar is more prominent.
- The ASL is presented as a direct translation of the visual form: a list of items; the calendar is not presented as a table.
- The ASL is essentially informative, although interaction is possible. The calendar
Figure 5. The user interface. A $3 \times 3$ grid of keys maps to 9 commands. Italicised text gives the output for each command if the date is currently set at February 28 2010; speech was kept deliberately terse.

is entirely interactive, as it serves no purpose if the user does not use it to navigate to and select a date.

5. Evaluation

The evaluation was completed as part of a wider experiment (Jay et al. 2010), the aim of which was to test whether the audio interfaces derived from the eye-tracking studies provided better access to dynamic content (including auto-suggest lists, slide shows, tabs, and others) than screen readers do at present. The study used a within-subjects design, in which all participants completed a series of tasks as part of a holiday booking process on the HCW Travel Company Website (specially designed for the evaluation), using the Firefox browser with two adapted versions of the Fire Vox screen reading extension. With respect to ASLs and pop-up calendars, these differed in that one version (the ‘SASWAT’ browser) used the audio interfaces described above, while the other (the ‘base case’ browser) required the user to complete the input box manually. The latter is similar to the current behaviour of many proprietary screen readers.

5.1 Method

The goal of the evaluation was to give participants an opportunity to compare the behaviour of both browsers whilst interacting with a variety of dynamic microcontent, so they could later specify which they preferred in a structured interview. As such, they completed the same tasks, on the same website, twice. To control for practice effects, half the participants completed the holiday booking tasks using the base case browser first, the other half used the SASWAT browser first. Although
### Table 1: Details of participants in the final evaluation of the SASWAT Web browser.

This table gives the nature and onset of their visual impairments (note that all participants who are marked as registered blind had some residual sight), and the assistive technology and Web browser normally used by them.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Impairment</th>
<th>Onset</th>
<th>Assistive technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1m</td>
<td>profoundly blind</td>
<td>adulthood</td>
<td>JAWS</td>
</tr>
<tr>
<td>P2m</td>
<td>profoundly blind</td>
<td>birth</td>
<td>JAWS</td>
</tr>
<tr>
<td>P3f</td>
<td>partially sighted</td>
<td>birth</td>
<td>ZoomText</td>
</tr>
<tr>
<td>P4f</td>
<td>registered blind</td>
<td>childhood</td>
<td>JAWS</td>
</tr>
<tr>
<td>P5f</td>
<td>profoundly blind</td>
<td>adulthood</td>
<td>JAWS</td>
</tr>
<tr>
<td>P6m</td>
<td>registered blind</td>
<td>birth</td>
<td>JAWS</td>
</tr>
<tr>
<td>P7m</td>
<td>profoundly blind</td>
<td>adulthood</td>
<td>JAWS</td>
</tr>
<tr>
<td>P8f</td>
<td>registered blind</td>
<td>adulthood</td>
<td>ZoomText (no audio)</td>
</tr>
<tr>
<td>P9m</td>
<td>profoundly blind</td>
<td>birth</td>
<td>JAWS</td>
</tr>
<tr>
<td>P10f</td>
<td>registered blind</td>
<td>adulthood</td>
<td>ZoomText (no audio)</td>
</tr>
<tr>
<td>P11m</td>
<td>partially sighted</td>
<td>adulthood</td>
<td>Enlarged fonts</td>
</tr>
<tr>
<td>P12f</td>
<td>registered blind</td>
<td>childhood</td>
<td>JAWS</td>
</tr>
</tbody>
</table>

The evaluation was run on a MacBook Pro running OS X 10.5.8 Leopard. The evaluation was conducted with the Mac system voice ‘Alex’ at the speaking rate ‘normal’. Audio output was provided by the on-board speakers. Participants used a separate keyboard to navigate and provide typed input.

#### 5.1.1 Participants

Twelve participants were recruited for the evaluation. P1m was a member of staff and p2m and p3m were students at the University of Manchester. These participants had provided feedback in informal iterative evaluation sessions during development, so had thus had experience of using an earlier version of the SASWAT Web browser. P4f, p5f, p6m, p7m and p11m were service users and p8f, p9m and p10f were volunteers/staff at either Macclesfield Eye Society or Walthew House Deaf Blind Services in Stockport. P12f was an IT teacher who worked with blind students. P1m, p2m, p3f, p6m and p12f were in the 20-45 age group; the remaining participants were over 45.

Three of the participants were partially sighted, four were registered blind with some residual vision and five were profoundly blind. All participants normally used Windows, and the majority browsed the Web using the screen reader JAWS (v. 10 or below). P1m used a standard desktop computer set-up with a large font size, but had had experience of using JAWS at college. Table 1 summarises the nature and onset of the visual impairments of participants and describes the computer set-up used by the participants. Most participants browsed the Web every day. P4f browsed the Web weekly and p10f monthly. P7m had only browsed the Web twice before, for a short period in college. Common reasons for Web browsing included work, study, shopping and email. All used the Microsoft Internet Explorer Web browser, except for p12f, who used Firefox.

#### 5.1.2 Procedure

Participants were shown how to use the browser and given the chance to practice using the navigation commands. They then completed the tasks with each browser, and answered questions about their experience in the structured interview. This

---

1Note that this is the speech synthesizer used by Fire Vox, and is distinct from the Voiceover screen reader.
interview asked users: to rate the ease of use of the system; if they would like to be able to access this type of information; which browser they found easiest to complete the task with; which browser they found hardest; if information available to sighted users should also be available to users with a visual disability, and; for any other comments. Each session was audio recorded and participants were encouraged to vocalise any problems and ask for help when necessary. The general nature of each type of dynamic content was explained before participants interacted with them. The information sheet and script used for the experiment (including the interview questions) and transcripts of all the session recordings are available to download with the evaluation Technical Report (Jay et al. 2010).

5.1.3 Tasks

The tasks involving ASLs and calendars were found on the booking page of the Web site. This page allowed participants to enter their destination and departure and return dates. A link was also present that, when selected, expanded a section of the page to display instructions. After viewing these instructions participants were given the following task:

Enter ‘Spain: Andalucia’ in the ‘Destination’ text field.

When the user typed in the ‘Destination’ text field, a list of suggestions for holiday locations appeared under the box matching the characters that had been entered so far. Users of the base case browser were unaware of this and entered text manually. Users of the SASWAT browser were presented with the instructions in the manner described above. The next task was to set the dates:

Set the departure date to the 1st January 2010, and the return date to the 2nd February 2010.

When the focus entered one of the date fields a pop-up calendar appeared. This was from the Dojo toolkit, styled in the form shown in Figure 2(b). Initially the current date was displayed in the date field (in the format dd/mm/yyyy), and this was highlighted in the calendar by a box. Users of the base case scenario were unaware of this; when the focus entered the date field they heard an announcement in the form: ‘departure date twenty six, one, two thousand and ten’. The date was set by typing. Users of the SASWAT browser heard the same announcement, then ‘date entry’, followed by the date again (in full, e.g, ‘Tuesday January twenty-six two thousand and ten’) to indicate that it was in a mode that allowed users to access the calendar. Users could then either type manually, or navigate and select a date as described above.

5.1.4 Interview

After they had completed the tasks with each browser, participants were asked for their opinions of how the two different browsers dealt with the various types of dynamic micro-content. For each item of dynamic micro-content, participants were asked whether they had encountered it before on the Web, to give their opinion on how easy it was to use, whether they would like to be able to access it, and whether they could think of any ways of improving access to it in audio. Finally, participants were asked which browser they preferred overall and why, whether it was useful to be notified when information on a Web page changed, and about their views on making content accessible.
5.2 Results

The evaluation showed that the audio implementations of both the ASL and the calendar were popular with the users. Both were, on the whole, preferred to manual input, and resulted in significantly fewer input errors. The detailed results for each are below, while Table 2 gives an overview of the errors made by participants when completing the form.

5.2.1 ASL

All participants except p3f (who said she found it easier just to type in the text field) said they would like to have access to them when browsing the Web. P4f commented that auto-suggest lists were ‘very important to a visually impaired person’, p2m described them as ‘useful’ and p10f said ‘they might be useful’. P8f said she would like to use it as an alternative to typing and thought the fact that an entry on the list was already correctly spelled and formatted made inputting a destination ‘much easier’.

In the base case browser, p12f pressed enter after typing ‘Spain’, which selected the first entry from the (visual) auto-suggest list. She was not able to hear the suggestion, and did not realise that this would happen. All other participants required help spelling or formatting the destination.

The participants were generally positive about the access provided by the SASWAT browser. P4f and p11m described it as ‘quite good’, whilst p6m and p8f described it as ‘very good’ and p12f ‘really good’. P5f was initially sceptical – ‘when you started talking about it, I thought what does she mean? And I’m thinking, oh god it sounds technical!’ – but after she had used the auto-suggest list thought it was ‘very useful actually’. P6m said it was ‘like using the predictive text on the phone’.

P7m was particularly enthusiastic about the auto-suggest list, thinking it was ‘brilliant – I love that. Great. That is really, really, really helpful... I thought that was brilliant, absolutely... it’s such an easy way of doing things.’ This may have been because he found it difficult to type the destination name: ‘The thing is, with somebody like myself, who’s totally blind, after a bit you get somewhat frustrated about not achieving what you want to achieve and getting where you want to be... ’Cause it does get frustrating. But with the information that you’ve just suggested there, with the helper on the second system, that’s brilliant to me. It just makes it so much easier to get to where you want to be.’

Suggestions for improvement included:

• Slowing the suggestions down.
• Differentiating items, e.g. saying ‘option 1’.
• Announcing the first item only, indicating the presence of further items with a non-speech sound.
• Using different keys to navigate the list.
• Having optional instructions on how to use the list.

5.2.2 Calendars

On the whole, participants found the calendar date entry easier to use than typing the date manually. When typing manually, all participants made either formatting or typing errors, or both, or asked the experimenter what format was needed. Only one third of participants (p1m, p2m, p4f and p11m) did not make any typing errors. It should be noted that all participants were using an unfamiliar keyboard, and that errors were often related to the special character separating the days, months and years.
Table 2. Errors made using the different interfaces for auto-suggest lists and date entry. Users had to enter ‘Spain: Andalucia’ into the ASL and two dates formatted as ‘dd/mm/yyyy’. An ‘H’ indicates that help was required, a ‘x’ that the value was entered with an error, and ‘✓’ that the value was correct when submitted. Error types are given in parentheses: ‘c’ indicates case did not match, ‘s’ a typing error (e.g., missing letter), ‘f’ that formatting was not correct. Errors that were corrected by participants before submitting the form are not noted here.

<table>
<thead>
<tr>
<th>Participant</th>
<th>ASLs</th>
<th>Calendars</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Manual entry</td>
<td>Audio interface</td>
</tr>
<tr>
<td>P1m</td>
<td>✓ H</td>
<td>✓</td>
</tr>
<tr>
<td>P2m</td>
<td>√ (s,c) H</td>
<td>✓</td>
</tr>
<tr>
<td>P3f</td>
<td>✓ H</td>
<td>✓</td>
</tr>
<tr>
<td>P4f</td>
<td>✗ H</td>
<td>✓</td>
</tr>
<tr>
<td>P5f</td>
<td>√ (c) H</td>
<td>✓</td>
</tr>
<tr>
<td>P6m</td>
<td>√ (c) H</td>
<td>✓</td>
</tr>
<tr>
<td>P7m</td>
<td>✓ H</td>
<td>✓</td>
</tr>
<tr>
<td>P8m</td>
<td>√ (c) H</td>
<td>✓</td>
</tr>
<tr>
<td>P9m</td>
<td>√ (c) H</td>
<td>✓</td>
</tr>
<tr>
<td>P10f</td>
<td>√ (f) H</td>
<td>✓</td>
</tr>
<tr>
<td>P11m</td>
<td>√ (f,c) H</td>
<td>✓</td>
</tr>
<tr>
<td>P12f</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

No participants conclusively determined the format to use: 4 of the 12 (p1m, p8f, p9m, p11m) immediately asked the experimenter what the format should be, while another three (p2m, p3f, p10f) asked indirectly. Three further participants (p5f, p6m, p7m) asked or were told after having difficulties or making errors. Even with such help from the experimenter, some participants submitted the form without the correct date in the correct format. When using the audio date selection tool, all participants correctly entered both dates, without error.

Positive comments about the calendar ranged from ‘alright, yeah’ (p4f) and ‘ace’ (p1m) to ‘great, that is brilliant, that is really, really good’ (p7m). Three participants (p10f, p2m, p1m) stated that they thought it important to still be able to enter the date manually, one of whom (p10f) said that this would be her preference. P3f, p5f, p6m, p7m, p8f and p12f all said that using the calendar was much easier than entering the date manually. Part of the reason for this may have been because using the keyboard to type the date was difficult: p6 commented specifically on the difficulty of moving between the number keys and the slash key. It was not only novice users who had trouble entering the date: p3f, p1m and p12f who use computers at work on a daily basis all found typing the date difficult and time-consuming.

Most of the participants wanted to have access to the calendar, but p9m felt it was ‘very fiddly’, and ‘I think I’d prefer to type it in, ’cause that’s quicker.’ P11m really liked using calendars when looking at the screen (he has partial sight), but thought he would prefer to type the date when using a screen reader, and was not convinced of the benefit of providing calendars in audio. It is worth noting, however, that all of the participants, including those who were sceptical about the need to use the calendar, either asked for help with formatting the date, or entered the date incorrectly (p9m, in fact, did both of these things). Three participants (p1m, p9m, p5f) suggested improving the implementation by changing the keys used.
6. Discussion

It is possible, and relatively simple, to present dynamic content in the same way as other Web content, by giving access to the document elements and allowing the user to move around them. These experiments show, however, that this is not necessarily the most effective approach, and can lead to accessibility that is only superficial. The eye-tracking studies showed that although auto-suggest lists and pop-up calendars are visually similar, they are viewed and used in quite different ways. While a simple audio translation of the list could bring some benefits of ASLs to the user, providing audio access to a table of dates to someone wishing to enter a date is unlikely to provide the benefits that it does visually.

It was noted in the introduction that these types of widget are essentially optional, in that their use is not critical to task completion. Indeed, we speculate that this is one important reason for the lack of handling of this type of update in screen readers. The eye-tracking study, however, showed that ASLs and pop-up calendars were almost always viewed and were more often used than not. This suggests that they do offer some advantage to these users, and this was clearly demonstrated in the evaluation. Both widgets make data entry much less prone to errors than when done manually. This was particularly notable for calendars, where uncertainty about the correct format and typing errors caused by needing special characters (the slashes or dashes used to separate days, months and years) make entering a date accurately a surprisingly difficult task. The importance of this assistance is actually probably greater for screen reader users, as it is more difficult and time-consuming for these people to review their input and to scan the surrounding page for clues (e.g., dd-mm-yy near a date entry field).

Our approach is to use eye-tracking, as this helps to build an understanding of how sighted users use, and benefit from, Web content. We then use this to formulate these benefits as requirements for an audio implementation. Finally, these requirements are used to design, from scratch, an interface that can be used in the context of a screen reader. Importantly, this design process includes iterative testing with screen reader users, to refine the user-interface and identify additional needs or problems associated with non-visual browsing.

Considering the two design patterns independently, the interfaces that resulted from the eye-tracking studies can be compared to the ARIA design patterns for ‘datepicker’ and ‘autocomplete’. When this is done, it can be seen that our experiments provide empirical support for the ‘datepicker’. With the autocomplete, however, the recommendations cover only keyboard interaction, not (explicitly) how the screen reader should present the information. Thus, our findings can be used to guide screen reader developers in how to handle such design patterns, in particular, we recommend automatically speaking the first three suggestions as they appear or change. Finally, it should be noted that the implementations did not require any ARIA markup (only access to the live DOM), so date selection using this type of interface could be applied by a screen reader to any input field identified as a date input field, and auto suggest lists could be presented as they appear on any site.

The two applications of this approach that have been described above result in quite different audio implementations. The audio ASL is similar to how a direct translation would appear, albeit with important subtleties that improve interactivity. The calendar, however, took on quite a different form, bearing little resemblance to the table viewed by sighted users. The evaluations showed that both achieved

---

1 As described at http://www.w3.org/TR/wai-aria-practices/#aria_ex
their aims: the benefits that sighted users gain from these widgets are brought to
the screen reader user in an audio form.

7. Experimental Resources

Technical reports describe the eye-tracking study (Jay and Brown 2008) and final
evaluation (Jay et al. 2010) in more detail. Available with these reports are associ-
ated files containing the eye tracking data, participant information sheets, consent
forms, questionnaires, and transcriptions.

8. Acknowledgements

This work is part of the Single Structured Accessibility Stream for Web 2.0
Access Technologies (SASWAT) project and is funded by the UK EPSRC
(EP/E062954/1). The authors would like to thank them for their continued sup-
port.

References

Proceedings of the 2010 Cross-Disciplinary Conference on Web Accessibility
of graph based diagrams. In: Proceedings of the 8th International Conference
Brown, A. and Jay, C., A Review of Assistive Technologies: Can users access dy-
namically updating information?. Technical Report, University of Manchester,
Guidelines for Audio Presentation of Graphs and Tables. In: ICAD 2003 Work-
shop on Auditory Displays In Assistive Technologies, Boston, MA, USA.
2007 international cross-disciplinary conference on Web accessibility (W4A)
Banff, Canada New York, NY, USA: ACM, 1–6.
havior in WWW search. In: SIGIR ’04: Proceedings of the 27th annual inter-
national ACM SIGIR conference on Research and development in information
retrieval Sheffield, United Kingdom New York, NY, USA: ACM, 478–479.
Jay, C. and Brown, A., User Review Document: Results of Initial Sighted and Visu-
ally Disabled User Investigations. Technical Report, University of Manchester,
Jay, C., Brown, A.J. and Harper, S., Internal evaluation of the SASWAT audio
browser: method, results and experimental materials. Technical Report, Uni-
versity of Manchester, Available from http://hcw-eprints.cs.man.ac.uk/125/,
2010.
Kildal, J. and Brewster, S., 2005. Explore the matrix: Browsing numerical data ta-


