The Design and Evaluation of an Auditory-Enhanced ScrollBar

Stephen A. Brewster, Peter C. Wright and Alistair D. N. Edwards

Department of Computer Science,
University of York,
Heslington,
York, Y01 5DD, UK.
Tel.: 0904 432765
sab@minster.york.ac.uk

ABSTRACT

A structured method is described for the analysis of interactions to identify situations where hidden information may exist and where non-speech sound might be used to overcome the associated problems. Interactions are considered in terms of events, status and modes to find any hidden information. This is then categorised in terms of the feedback needed to present it. An auditory-enhanced scrollbar, based on the method described, was then experimentally tested. Timing and error rates were used along with subjective measures of workload. Results from the experiment show a significant reduction in time to complete one task, a decrease in the mental effort required and an overall preference for the auditory-enhanced scrollbar.

KEYWORDS

Auditory interfaces, multi-modal interfaces, earcons, sonification, auditory-enhanced widgets

INTRODUCTION

It has been shown that non-speech sounds can be designed to communicate complex information at the human-computer interface [3, 4, 8]. Unless these sounds supply information that users need to know they will serve no real purpose; they will become an annoying intrusion that users will want to turn off. Therefore an important question that must be answered is: Where should non-speech sound be used to best effect in the graphical human-computer interface?

The combination of graphical and auditory information at the interface is a natural step. In everyday life both senses combine to give complementary information about the world; they are interdependent. The visual system gives us detailed data about a small area of focus whereas the auditory system provides general data from all around, alerting us to things we cannot see. The combination of these two senses gives much of the information we need about our environment. These advantages can be brought to the human-computer interface. Whilst directing our visual attention to one specific task, such as editing a document, we can still monitor the state of other tasks on our machine. At the present time almost all information presented goes through the visual channel. This means information can be missed because of visual overload or because the user was not looking in the right place at the right time. An interface that integrates information output to both senses can capitalise on the interdependence between them and present information in the most efficient way possible.

The work reported here is part of a research project looking at the best ways to integrate auditory and graphical information at the interface. The sounds used in this paper are based around structured audio messages called *Earcons* first put forward by Blattner, Sumikawa & Greenberg [3] and experimentally tested by Brewster, Wright & Edwards [4]. Earcons are abstract, synthetic tones that can be used in structured combinations to create sound messages to represent parts of an interface.

One question that might be asked is: Why use sound to present the extra information? A graphical method could be used instead. The drawback with this is that it puts an even greater load on the visual channel. Furthermore, sound has certain advantages. For example, it can be heard from 360° all around, it does not disrupt the user’s visual attention and it can alert the user to changes very effectively. It is for these reasons that we suggest that sound be used to enhance the user interface.

How should the information be apportioned to each of the senses? Sounds can do more than simply indicate errors or supply redundant feedback for what is already available on the graphical display. They should be used to present information that is not currently displayed (give more information) or present existing information in a more effective way so that users can deal with it more efficiently. Alty [1] has begun considering this problem in process control systems. Sound is often used in an *ad hoc* way by individual designers. In many cases no attempt is made to evaluate the impact of sound. Gaver [8] used a more principled approach in the SonicFinder that uses sounds in ways suggested by the natural environment. For example, selecting an item gave a tapping sound or dragging gave a scraping sound. However, in an interface there are many situations where there are no natural equivalents in the everyday world. The SonicFinder used sound redundantly with graphics. This proved to be very effective but we suggest that sound can do more. A method is needed to find situations in the interface where sound might be useful but currently there is no such method. It should provide for a clear, consistent and effective use of non-speech audio across the interface. Designers will then have a technique for identifying where sound would be useful and for using it in a more structured way rather than it just being an *ad hoc* decision.

This paper describes a structured, informal method of analysing the interface to find hidden information that can cause user errors. It then describes an experiment to test an auditory-enhanced scrollbar designed, using the analysis technique, to overcome the problems brought to light. The aim of the experiment was to justify the model and see if auditory-enhanced widgets were effective.

Workload Assessment

Wright & Monk [14] have argued that quantitative measures of usability such as error rates and performance times do not provide a complete picture of the usability problems of an interface. They argue that users may perform tasks well and quickly and yet find the system frustrating to use and that it requires more effort to complete a task than they would expect. This dissociation between behavioural measures and subjective experience has also been addressed in studies of workload. Hart and Wickens [11] for example, define workload "...as the effort invested by the human operator into task performance; workload arises from the interaction between a particular and task and the performer". Their basic assumption is that cognitive resources are required for a task and there is a finite amount of these. As a task becomes more difficult, the same level of performance can be achieved but only by the investment of more resources. We used a workload test as part of our usability evaluation of the auditory-enhanced scrollbar discussed in this paper. It provides a more rounded and sensitive view of usability than just time and error analysis alone.

WHERE TO USE SOUND AT THE INTERFACE

There are potentially many ways that sound could be used in an interface This paper suggests using it to present information that is hidden from the user. This is a similar approach to that taken by Blattner, Papp & Glinert [2] when adding sound to maps. Here information is hidden because of visual clutter. Information may be hidden for a number of reasons:

Information is not available: It may not be available on the display due to hardware limitations such as lack of CPU power or screen size.

Information is difficult to access: Information may be displayed but be difficult to get at. For example, to get file size and creation date information on the Macintosh a dialog box must be called up.

Small area of visual focus: Information can be hidden because it is outside the small area of focus of the visual system. The user may not be looking at the right place at the right time to see what is displayed. One of the advantages of sound is that it can be heard from 360˚ all around without having to take visual attention away from the current task.

Screen space: There is a trade-off between screen space and salience of status information [12]. In order for status information to be visually conspicuous it must take up much screen space. This leaves less for what the user is working on so that some will be hidden. An advantage of sound is that it takes up no screen space.

Modes: A mode is a state within a system in which a certain interpretation is placed on information (for example in one mode characters typed into an editor may appear on the screen, in another they may be interpreted as commands). It is often the case that the details of this interpretation are hidden from the user and mode errors result [13].

Dix *et al.* [5] suggests two principles that, if satisfied, could reduce the number of errors due to hidden information: *Observability* and *predictability*. The user should be able to predict what the effects of their commands will be by observing the current status of the system. If information about the system state is hidden, and not observable, then errors will result. The above points show that hidden information is a problem and we suggests using sound to overcome it. A method is needed to find out where such hidden information exists.

Event, Status and Mode Analysis

Analysing interactions with event and status information was first put forward by Dix [6]. This has been extended to include explicit information about modes, one of the most important areas of hidden information in the interface. A description of each of the types of information follows.

Event: A discrete point in time. It marks an important happening in the system. An event can be initiated by the user (mouse click) or system (mail arriving). There can be input events (mouse clicks) or output events (a beep indicating an error).

Status: This is information about the state of the system that is perceivable by the user. If there is information about the state that is not in the displayed as status information then it will be hidden from the user. Status refers to something that always has a value such as an indicator. It is relatively static and continues over time. The rendering is how the information about the state of the system is displayed. Thus status information can be rendered in different ways, for example graphically, through sound or a combination of both.

Mode: This maps the events onto operations on the system state. Different modes can make the same events have different effects. If the mode is not displayed then the user may not know the mapping.

Problems occur when:

• Events are not saliently signalled to the user, for example the user does not notice when an item of mail arrives.

• Events change the rendering but do not give adequate feedback, for example dragging the scrollbar thumb where one can lose it without noticing.

• Events that change the state are not adequately reflected in the rendering, for example a mode change in the VI text editor. Here the change in state from command to insert mode (and vice versa) is not reflected in the rendering at all. There is no way to observe which mode the system is in.

Characterisation of the Feedback

Once the event, status and mode information has been extracted it must be characterised in terms of the feedback needed to present it. The method described here is based around that of Sellen, Kurtenbach & Buxton [13]. There are four dimensions of feedback:

Action-dependent versus independent: Does the feedback depend on user/system actions taking place? Events are

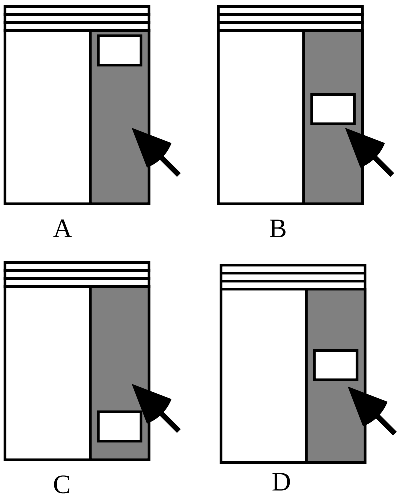


Figure 1: Scrollbar 'kangarooing'.

action dependent; an action on the part of the user or the system must occur for an event to take place. Status feedback is independent of activity; it continues over time whether there are actions or not.

Transient versus sustained: How long does the feedback last? Events are transient, they occur at momentary, discrete points. Status information is sustained and continues over time.

Static versus dynamic: Does the feedback change when it is presented or is it constant? Status information can be static, for example a window onto a file directory, or dynamic and change over time, for example a CPU load indicator. Events are static; they only occur for a short time and indicate that one particular thing has happened.

Demanding versus avoidable: Should the user be forced to perceive the feedback or not? Events are usually demanding as they are important occurrences. Status information should be avoidable. It exists all the time and the user can sample it only if necessary. This is not necessarily the case as the user may not be interested in some events (for example, the arrival of some types of mail) and may not want to miss some types of status information (for example, they may want to monitor a compilation and act when it has finished). This aspect of the categorisation can capture the urgency of the information to be presented. The work by Edworthy *et al.* [7] can be used to make sounds demanding (more urgent) or avoidable (less urgent) as required.

Event, Status and Mode Analysis Applied

This informal, structured technique can be used to investigate problems with interactions and find what might be causing the mistakes. The technique has been used to analyse buttons, alert boxes, menus and windows. As an example, this paper will discuss the problems of scrollbars. As Myers' [11] shows, scrollbars vary little in their general design. Therefore this analysis will be applicable to many different scrollbars. There are three common problems with scrollbars:

Dragging the thumb wheel out of the 'hot spot': The thumb wheel is the part of the scrollbar that allows the user to move by an arbitrary amount (the white box in Figure 1). When dragging the thumb up or down in the document one may move too far to either the top, bottom, left or right of the scroll bar and the thumb will be lost. The document will then not scroll and the user may become confused. When dragging only the outline of the thumb is moved which does not grab the user's attention.

Position awareness in the document: When scrolling through a document it can be hard to maintain a sense of position. The text can scroll by too fast to see and the thumb only gives an approximate position. Some systems such as Microsoft Word™ put a page count in the bottom left hand corner of the screen but this is too far from the centre of visual focus so is hidden. One other method, used by MacWrite™, is to put the page number in the thumb wheel itself. This is closer to the user's centre of visual focus and therefore should be more effective. The problem with this is that the thumb is very small so only a small amount of data can be displayed. It may also be missed by the user if they are looking at the main part of the window and not the scroll bar. It may force the user to look at the scrollbar when they really want to look at the screen.

'Kangarooing' with the thumb wheel: Repeatedly clicking in the area above or below the thumb scrolls by a window-sized step. When the thumb reaches the target position (pointer location) it will scroll down to the window below and then back up to the window above the target and keep on doing this until the user notices and stops clicking. If the document is scrolling fast then it can be hard to tell this is happening. Figure 1 shows an example. In A the user begins to scroll down towards the pointer. In B the thumb wheel is just above the pointer. In C the user has clicked and scrolled below the pointer. In D the user clicked again and the thumb scrolled back above the pointer so kangarooing occurred. Unless the user is looking at the thumb it can be hard to spot that this has happened.

Let us consider the problems of position and kangarooing as examples and see if some of the errors can be identified. To use the Event, Status and Mode (ESM) technique first think of the 'generic' scrollbar which would provide all the information required and where nothing was hidden. Identify all the event, status and mode information and then the feedback required to present this. Then do the same for the real scroll bar, identifying the information actually present. If there is less in the real scroll bar than the generic one then that is where hidden information exists. If there is more, then there may be redundant information that could be removed. This is similar to the approach taken by Kishi [10] where a 'model' or expert user's interactions are compared to those of normal users and where the differences occur are usability problems.

Figure 2 illustrates the method for using the ESM analysis technique. The three types of information must be identified by looking at the scrollbar and considering the feedback present with the descriptions of events, status and modes supplied above; then see which parts of the feedback fit into which category of information.

Generic scrollbar: There are two events. The first is the user clicking in the window scroll area and the second is the thumb reaching the target location. Once the information has been extracted then it must be categorised. Feedback from the click is demanding as the user has actively to press the mouse button; it is static as it does not change; it is action-dependent: The user must press the button; and it is transient: The click only lasts a short time. When the thumb reaches the target a demanding event is given so that the user knows that they are where they want to be. The rest of the feedback is similar to before. There are three types of status information: Information from the thumb and when it moves; information from what is in the window and when that changes; and position in document indication. This may be a page count in the thumb or just where the thumb is in the scrollbar. Thumb movement feedback is demanding so that the user knows when it has happened; it is dynamic because the feedback changes: The thumb moves; it is sustained as the thumb is always visible; and it is action dependent: It only moves when clicked by the user. Window scrolling and position in document are similar. In this example there are no real modes. When the mouse button is down the system is in a mode and the user gets demanding feedback, as mentioned above, because they actively have to press the button.

Real scrollbar: There is no event to indicate that the thumb has reached the target, which is why kangarooing can occur. The status information is the same as above but thumb movement feedback is avoidable: It is easy to avoid seeing the thumb move as it is small. The same is true of a position in document indicator that again is avoidable as it is out of the area of visual focus. Again, there are no hidden mode problems in the real scrollbar. Some potential reasons for the problems with scrollbars are thus demonstrated. The sounds added to overcome these are described in the Sounds Used section below. The analysis of some of the problems associated with a scrollbar has shown the ESM technique in practice. We need to find out if such a scrollbar was built whether it would be more usable.

EXPERIMENT

An experiment was designed to test an auditory-enhanced scrollbar based on the design described above. The aim of the experiment was to examine the ESM method to see if the changes it suggested would improve usability. The experiment was a two-condition within-subjects design. In one half of the test subjects were given a standard visual scrollbar and in the other subjects were given an auditory-enhanced one (see Figure 3).

Subjects

The twelve subjects used were postgraduate students from the Department of Computer Science at the University of York. They all had more than three years experience of graphical interfaces and scrollbars.

Subjects	Condition 1	Condition 2
Six Subjects	Auditory Scrollbar Train & Test	Visual Scrollbar Train & Test
Six Subjects	Visual Scrollbar Train & Test	Auditory Scrollbar Train & Test

Figure 3: Format of the Experiment.

Tasks

Subjects were given two types of task. The first, which we will call the 'Search' Tasks, involved the subjects visually searching through a file of data to find significant features. These features were such things as whole line of 'a's together. When the target was found the subjects had to say which page the target occurred on. The other tasks, which we will call the 'Navigate' Tasks, involved subjects being given instructions to go to a specific point on a specific page and read the data that was there. For example, subjects were asked to go to page seven and read the first six characters of the first line. Along with these absolute navigation tasks relative tasks were also given. For example, subjects were asked to go up four pages from the current page and read the first six characters of the last line. The data were described to the subjects as 'experimental results data'. The rationale given to the subjects for the tasks was that they were searching through the data to find significant features for analysis.

Sounds Used

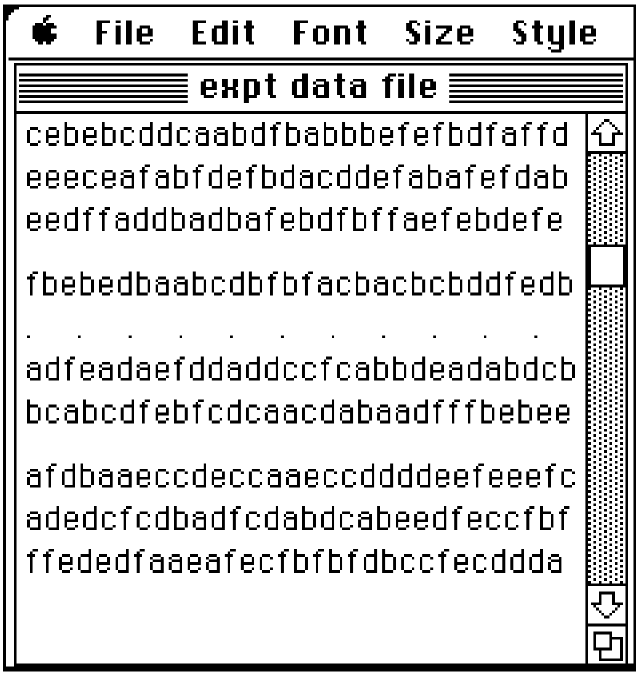
The earcons were designed using the guidelines put forward by Brewster *et al.* [4]. There were two sounds in the auditory-enhanced scrollbar:

Window scrolling/thumb movement: A fixed tone of duration 9/60^{ths} of a second was used to indicate a window scroll event. When the subject scrolled towards the bottom of the document a low-pitched note, 'C' below Middle 'C' (130Hz), was played. When scrolling up by a window a high-pitched note 'C' 3 octaves above Middle 'C' (2093Hz) was played. If the subject was scrolling downwards towards a target location they would hear the low-pitched sound. If kangarooing occurred then the subject would hear a demanding high-pitched tone when they did not expect it.

Page scrolling/position: A low intensity continuous tone gave status information about the current page. This was increased in volume for 9/60^{ths} of a second when a new page boundary was encountered (page boundary event) to demand the listener's attention. It then decreased again to just above threshold level so that it could be habituated. The

Generic Scroll bar: **Event** repeatedly click in window scroll area thumb reaches target **Status** thumb moves window scrolls position in document indication **Mode** when mouse button down	**Feedback:** **Event** *click*: demanding, static, dependent, transient *thumb reaches target*: demanding, static, transient, dependent **Status** *thumb movement*: demanding, dynamic, sustained, dependent. *window scroll*: demanding, dynamic, sustained, dependent *position*: demanding, dynamic, sustained, independent **Mode** mouse down in scrollbar indicates mode: demanding, sustained, dependent, static
Real Scroll bar: **Event** repeatedly click in grey area **No thumb reaches target event** **Status** thumb moves window scrolls position in document indication **Mode** when mouse button down	**Feedback:** **Event** *click*: demanding, static, dependent, transient **Status** *thumb movement*: **avoidable**, dynamic, sustained, dependent. *window scroll*: demanding, dynamic, sustained, dependent *position*: **avoidable**, independent, dynamic, sustained **Mode** mouse down in scrollbar indicates mode: demanding, sustained, dependent, static

Figure 2: ESM analysis of scroll bar loss of sense of position and 'kangarooing'



```
cebebcddcaabdfbabbbefefbdfaffd
eeeceafabfdefbdacddefabafefdab
eedffaddbadbafebdfbffaefebdefe

fbebedbaabcdbfbfacbacbcbddfedb
.  .  .  .  .  .  .  .  .  .  .
adfeadaefddaddccfcabbdeadabdcb
bcabcdfebfcdcaacdabaadfffbebee

afdbaaeccdeccaaeccddddeefeeefc
adedcfcdbadfcdabdcabeedfeccfbf
ffededfaaeafecfbfbfdbccfecddda
```

Figure 4: The browser program (reduced in size) showing example data and a page boundary marked by dots.

notes played when scrolling towards the bottom of the document decreased in pitch from ‘B’ two octaves above Middle ‘C’ (1975Hz) to ‘C’ an octave below Middle ‘C’ (130Hz). The notes played cycled through the scale of ‘C’ major. So, for example, when scrolling down from the top of the document, the first note played would be ‘B’ (1975Hz), then ‘A’, ‘G’, ‘F’, ‘E’, ‘D’ and on to ‘C’ of the octave below. The reverse occurred when scrolling up from the bottom of the document.

When the scrollbar was clicked the window sound was played first followed by the page sound after a 9/60$^{\text{ths}}$ of a second delay. All the sounds used were based around the electronic organ timbre on a Roland D110 synthesiser. They were controlled by an Apple Macintosh via MIDI through a Yamaha DMP 11 mixer and presented by loudspeakers.

Workload Testing

[9] analyse workload into six different factors: Mental demand, physical demand, time pressure, effort expended, performance level achieved and frustration experienced. They have developed a measurement tool, the NASA-Task Load Index (TLX) for estimating these subjective factors. This has been tested in a variety of experimental tasks. Workload measures are little used in the field of interface evaluation yet the six factors identified in TLX would appear to bear directly on usability. Thus it would seem valuable to incorporate an estimate of workload into out evaluation of the auditory widgets. We do this by using the NASA-TLX tool but we add a seventh factor: Annoyance. This is often quoted as a reason for not using sound in an interface [4] as it is argued that continued presentation of sound can be an annoyance for the user. In addition to these seven factors we also asked our subjects to indicate, overall, which of the two interfaces they felt made the task easiest.

Experimental Design and Procedure

The experiment was in two halves. Half of the subjects had the visual condition first the other half had the auditory condition first to counterbalance any learning effects (see Figure 3). After the first condition subjects filled in a set of TLX workload charts. After the second condition the subjects filled in the same set of charts plus some overall ratings. Instructions were read from a prepared script. A simple document browser was created on an Apple Macintosh, based around TinyEdit, an example program supplied by Symantec with Think Pascal™ (see Figure 4). This browser allowed subjects to navigate around a document using a scrollbar and indicated page boundaries with a dotted line, in a similar way to many wordprocessors. The scrollbar used in the browser only allowed clicking in the grey region above or below the thumb wheel to scroll by a window of data either way. The subjects could not drag the thumb wheel or scroll by lines using the arrows.

The data files used were made up of groups of three lines of thirty randomly generated ‘a’ to ‘f’ characters separated by a blank line. The test files had twelve pages of data where pages were 50 lines long and windows 33 lines long. Therefore scrolling by a window did not necessarily mean that a new page boundary would be reached each time. The data was displayed in 12 point Geneva font.

Visual Condition: Subjects used an ordinary Macintosh scrollbar. Training was given before the main test was started. The experimental procedure was described and then sample Search and Navigate tasks were undertaken using a training data file. In the main test subjects were given a task, when they were ready to start they pressed ⌘Y that started a timer. When they completed their task they pressed ⌘Y again, the timer was turned off and the time recorded. Other errors were recorded by the experimenter.

Auditory Condition: The audio-enhanced scrollbar described above was used. In the initial training of subjects for this condition the feedback provided by the scrollbar was described in detail. The training and testing then proceeded as described above for the visual condition.

Experimental Hypotheses

The hypotheses were based around the predictions of the ESM analysis technique described above. If there was more useful feedback from the widget then time to complete the tasks and error rates should be reduced. The model suggested that there should be fewer kangaroo-type errors as subjects would notice thumb movement in the auditory condition. Subjects should be able to maintain their sense of position in the document with more page feedback and therefore give fewer wrong page answers. The workload felt by subjects should be reduced as the extra feedback would provide information that the subjects need. Physical demand and time pressure will be unaffected as they were unchanged across conditions and were left in for completeness. There will be no increased frustration or annoyance due to the addition of sound as the auditory feedback will provide information that the subjects need.

RESULTS

TLX Results

Figure 5 shows the average scores for each of the workload categories plus our two extra ones. Each was marked in the range 0-20. T-tests were carried out on the auditory versus visual conditions for each of the workload categories. Mental demand was assigned the highest mark of all the workload scores indicating that the experimental task itself was difficult. However, mental demand showed a significant decrease for the auditory condition over the visual (T(11)=3.23, p=0.008). 9/12 subjects rated the auditory condition lower in effort than the visual but this failed to reach significance (T(11)=1.83, p=0.09). There were no significant differences in any of the other workload categories except for overall preference. In this case, the subjects were asked to rate which scrollbar made the task the easiest. Here the auditory scrollbar was significantly better than the visual one (T(11)=2.55, p=0.02).

Timing and Error Results

Along with workload tests, more conventional measures of time and error rates were taken. Figure 6 shows the total times taken by each of the subjects in the two conditions for the Search Tasks. 9/12 of the subjects performed faster in the auditory condition but there were no significant differences in time scores. However, an F-test between the auditory and visual conditions across subjects showed a significant reduction in the variance in the auditory condition (F(11,11)=3.98, p=0.05). Two kinds of errors were recorded: Kangaroo errors and wrong-page errors (see Events, status and mode analysis applied section for details). There were no significant differences in either of the error rates between the two conditions.

Figure 6 shows the total times for the two conditions in the Navigate Tasks. In these tasks there was a significant difference between the times taken. A paired T-test showed the auditory one was significantly faster than the visual (T(11)=2.20, p=0.05). As before, there was also a significant reduction in the variance in the auditory condition (F(11,11)=5.43, p=0.05). Again, there were no significant differences in the error rates between the two conditions. However, there was a reduction in both categories of error in this task. For example, the number of wrong-page errors fell from 51 to 40 in the auditory condition but this failed to reach significance.

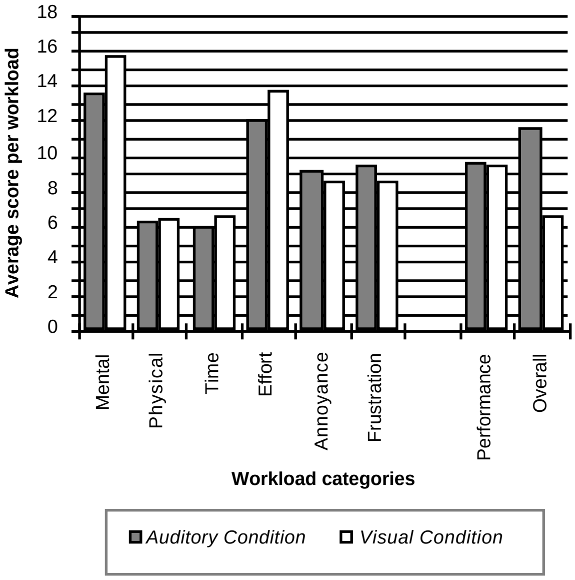


Figure 5: Average TLX workload scores for the Auditory and Visual conditions.

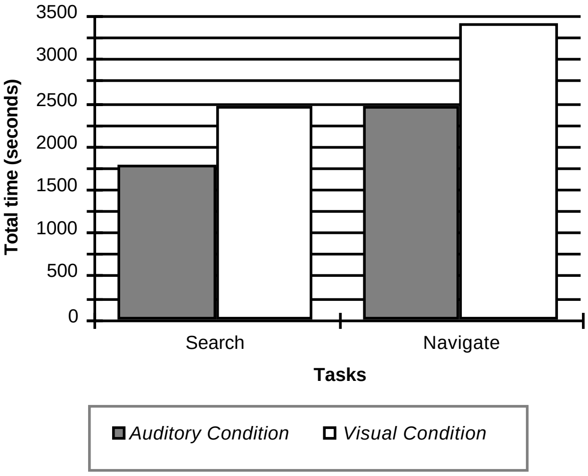


Figure 6: Total times for the Search and Navigate Tasks.

DISCUSSION

The workload results indicate that the auditory scrollbar reduced the workload of the task. Mental demand (which dealt with how much mental and perceptual activity was required) was significantly reduced. This could be due to it being easier for subjects to hear page boundaries than it was to see them as the feedback was more demanding. Although subjects felt their performance was no better in the auditory condition than in the visual, they had an overall preference for the auditory scrollbar because it lowered mental demand and there was some decrease in effort expended. These factors indicate that an auditory enhanced scrollbar would be an effective addition to an interface and could lower the workload therefore freeing up cognitive resources for other tasks. Workload measures are also shown to be an effective method of testing the usability of widgets.

The significant reduction in time for the auditory condition in the Navigate Tasks indicates that the auditory enhanced scrollbar improved performance. This is again evidence to suggest that auditory scrollbars are an effective extension to standard visual ones. The times for the Search Tasks were not significantly different. This may be due to the nature of the task. The subject was required to search visually through the data file and find a target. The advantage conferred by sound was lost in the overall time to do the visual searching. This visual searching took up a large proportion of the time for this task and the position awareness within the document was bound up in this. The advantages due to sound were small and therefore lost in the large times for visual searching. In the Navigate Task, where the subjects had to find a specific page searching was based on page boundaries so there was a better comparison between the auditory and visual conditions.

There were no significant differences in terms of errors between the two conditions. A problem with the error analysis was that the frequency of errors was too low to be a good measure. For example, in the Search Tasks there was less than one error per subject in each of the tasks. It turned out to be very difficult to generate many kangaroo type errors. It could be that as our subjects were experienced scrollbar users they had developed strategies for avoiding kangarooing in their everyday work that they used in the experiment. However, two subjects did say that the window scrolling sound did help them identify when a kangaroo event had taken place. Again, there were no differences between the condition for wrong-page errors. It may have been that subjects counted the page boundaries whether they saw them or heard them, but it just took longer when they have to do it visually. This may have been one of the reasons for improved performance in the Navigate Tasks for the auditory condition. Further investigation of errors is therefore necessary.

It is noteworthy that there were significant differences between the auditory and visual conditions in terms of variance on both tasks. Eight of the twelve subjects showed less variability in the auditory condition. However, a Sign test between conditions across subjects failed to reach significance. There is an indication that the variability has been reduced and further experiments would be needed to investigate this further.

Justification of the Event Status and Mode Method

The auditory-enhanced scrollbar was designed to overcome some of the problems identified by the ESM analysis. Do the experimental results justify the model? The addition of sound produced a significant improvement in performance on one of the groups of tasks and a decrease in the overall variability on both tasks. The mental workload required to perform the task was significantly less when sound was used and overall preference was for the auditory scrollbar. All these results indicate that the addition of sound was successful and the ESM model proven to be effective. One area that needs further investigation is that of error rates. The model predicts that the number of both kangarooing and wrong-page errors should be lower but the results failed to demonstrate this because not enough errors were generated in the experiment.

FUTURE WORK

In addition to giving information about page boundaries other events could be indicated when scrolling. In a programming editor, for example, events such as when a new procedure or function is reached could be displayed in sound. The scrollbar only allows sounds for 21 pages. This could be extended by using different rhythms or intensities along with pitch. Further widgets will be designed using the ESM analysis method. Currently we are testing audio enhanced buttons and windows. The experimental design described in this paper is being used as a testing framework. Further widgets will be tested in a similar two-condition within-subjects design and time, error and workload data will be collected.

CONCLUSIONS

An auditory-enhanced scrollbar was tested and found to significantly improve performance time on certain tasks. It also significantly reduced the mental workload and was rated with a significantly higher preference score than a standard visual scrollbar. This indicates that the integration of auditory feedback into graphical widgets is likely to provide more usable interfaces. The use of workload tests is also shown to be a useful way of measuring usability. The results from the experiment suggest that the event, status and mode analysis technique is effective in identifying areas in the interface where problems occur. Until now there was no structured approach to adding sound to an interface, it was done in an *ad hoc* way by individual designers. The method described provides a way of integrating sound in a consistent and effective way to produce a more usable interface. The results of the work described here show that widgets that combine both auditory and visual feedback are more effective as they make use of the natural way that humans deal with information in everyday life.

This work is supported by SERC studentship 90310837.

REFERENCES

1. Alty, J.L. & McCartney, C.D.C. (1991). Design of a multi-media presentation system for a process control environment. In *Eurographics multimedia workshop, Session 8: Systems,* Stockholm.

2. Blattner, M., Papp, A. & Glinert, E. (1992). Sonic Enhancements of two-dimensional graphic displays. In G. Kramer (Ed.), *Auditory Display: The Proceedings of the First International Conference on Auditory Display.,* Santa Fé: Addison-Wesley (In Press).

3. Blattner, M., Sumikawa, D. & Greenberg, R. (1989). Earcons and icons: Their structure and common design principles. *Human Computer Interaction*, 4(1), pp 11-44.

4. Brewster, S.A., Wright, P.C. & Edwards, A.D.N. (1992). A detailed investigation into the effectiveness of earcons. In G. Kramer (Ed.), *Auditory Display: The Proceedings of the First International Conference on Auditory Display.,* Santa Fé: Addison-Wesley (In Press).

5. Dix, A., Finlay, J., Abowd, G. & Beale, R. (1993). Status/Event Analysis. In *Human-Computer Interaction*, pp. 325-334. London: Prentice-Hall.

6. Dix, A.J. (1991). Chapter 10: Events and Status. In *Formal Methods for Interactive Systems*, pp. 239-270. London: Academic Press.

7. Edworthy, J., Loxley, S. & Dennis, I. (1991). Improving auditory warning design: Relationships between warning sound parameters and perceived urgency. *Human Factors*, 33(2), pp 205-231.

8. Gaver, W. (1989). The SonicFinder: an interface that uses auditory icons. *Human Computer Interaction*, 4(1), pp 67-94.

9. NASA Human Performance Research Group. (1987). *Task Load Index (NASA-TLX) v1.0 computerised version* NASA Ames Research Centre.

10. Kishi, N. (1992). SimUI: Graphical user interface evaluation using playback. In *Proceedings of the Sixteenth Annual International Computer Software & Applications Conference*, Chicago, Illinois: IEEE Computer Society, pp 121-127.

11. Myers, B. (1990). All the widgets. *ACM SIGRAPH Video Review*, CHI'90 Special Issue (57).

12. Scott, D. (1993). Status conspicuity, peripheral vision and text editing. *Behaviour and Information Technology*, 12(1), pp 23-31.

13. Sellen, A., Kurtenbach, G. & Buxton, W. (1992). The prevention of mode errors through sensory feedback. *Human Computer Interaction*, 7, pp 141-164.

14. Wright, P.C. & Monk, A.F. (1989). Evaluation for design. In A. Sutcliffe & L. Macaulay (Eds.), *People and computers 5*, pp. 345-358. Cambridge: Cambridge University Press.