

Correcting Menu Usability Problems with Sound

Stephen A. Brewster and Murray G. Crease

Glasgow Interactive Systems Group

Department of Computing Science,

University of Glasgow,

Glasgow, G12 8RZ, UK.

Tel: +44 (0)141 330 4966 / Fax: +44 (0) 141 330 4913

{stephen, murray}@dcs.gla.ac.uk

www.dcs.gla.ac.uk/~stephen/

Abstract

Future human-computer interfaces will use more than just graphical output to display information. In this paper we suggest that sound and graphics together can be used to improve interaction. We describe an experiment to improve the usability of standard graphical menus by the addition of sound. One common difficulty is slipping off a menu item by mistake when trying to select it. One of the causes of this is insufficient feedback. We designed and experimentally evaluated a new set of menus with much more salient audio feedback to solve this problem. The results from the experiment showed a significant reduction in the subjective effort required to use the new sonically-enhanced menus along with significantly reduced error recovery times. A significantly larger number of errors were also corrected with sound.

1. Introduction

Human-computer interfaces currently rely almost entirely on graphical feedback to present information to the user. However, research is now showing that sound combined with graphics can significantly improve usability by taking advantage of our natural ability to share tasks across sensory modalities, for example Alty (1995), Brewster *et al.*

(1995a), Beaudouin-Lafon and Conversy (1996). Such multimodal interfaces allow a greater and more natural communication between the computer and the user. They also allow the user to employ appropriate sensory modalities to solve a problem, rather than just using one modality (usually vision) to solve all problems. We believe that for these reasons sound should become a standard, integrated component in human-computer interfaces of the future. One question that remains is how should sound be used to improve usability?

This paper describes an experiment to investigate the integration of sound into standard graphical menus to overcome usability problems. Menus are a standard feature of most graphical interfaces because they are a convenient way of grouping commands to improve accessibility. As Hall and Bescos (1995: 384) say 'not only are menu driven interfaces easy to implement, but also considerable functionality can be derived from their use'.

Norman (1991) has a detailed discussion of all aspects of menus and their use. The design of menus has become almost standard across systems. However, they are not without problems. One of these is the mis-selection of menu items due to inadequate feedback. Norman has very little discussion of the feedback required to indicate a selection. He does say (p 157) 'the system should provide feedback to the users confirming selection and informing them of its progress'. He does not suggest how this should be done. It is normally assumed that the visual highlighting of the menu item is sufficient. We believe that this is not the case. In the rest of this paper we discuss the problems with menus, how they might be solved with more salient feedback and then describe an experiment to discover if adding sound improves usability.

1.1 The problem with menus

Pull-down menus are operated by pressing the mouse button down in the menu bar (or on a hot-spot to bring up a pop-up menu). On some systems the mouse button must be held down so that the menu remains visible, on other systems the menu may stay pulled down until the user presses the mouse button again (this paper concentrates on the former but both types of menu suffer from the problems described here). Moving the cursor along the menu bar highlights the menu title the cursor is in (usually with a visual effect such as reverse video) and displays the items in that menu. When the cursor is moved into the menu the items are highlighted as the mouse moves over them (unless the item is a divider or greyed-out). If the cursor is moved out of the menu the menu remains displayed but the highlight is removed. This provides a back-out facility in case the user decides that no menu item is to be chosen. If the cursor is over a valid item when the button is released that item is selected and then the menu is removed. On

some systems (e.g. the Macintosh) the user can set the item selected to flash between zero and three times to indicate that it has been selected before the menu is removed. If the button is released when the cursor is not over a valid menu item, the menu is simply removed. This behaviour is described in more detail by Crease (1996).

A study of menu interactions showed three possible outcomes of a selection:

- A correct selection is made.
- An incorrect selection is made.
- No selection is made.

An incorrect selection could happen for one of two reasons:

- The wrong menu item was mistakenly selected - a *mis-selection*. In this case the user simply chose the wrong item.
- The cursor slipped onto the wrong menu item accidentally as the button was being released - an *item slip*. Our study of menus showed that, sometimes, just as the mouse button was released users could slip into an adjacent menu item. This happened partly because the action of releasing the mouse button could move the mouse a little and also because users often started to move the mouse to the location of the next action before the mouse button was released. Because the menu is removed immediately after an item has been chosen this can be very hard to see.

Similarly, no selection could happen for one of three reasons:

- The user decided not to select anything and moved the cursor out of the menu before releasing the button.
- The cursor slipped off the edge of the menu as the button was being released - a *menu slip*.
- The user selected a disabled item or a divider. This could happen as the result of an item slip or the user deciding not to select anything.

Examples of a correct selection, an item slip and a menu slip can be seen in Figure 1. Menu and item slips reduce the usability of menus and can sometimes cause more serious problems. For example, if a user slipped off when trying to save and did not notice, then his/her data would not be saved with perhaps serious consequences. We therefore decided to investigate these problems further to see if menus could be improved.

FIGURE 1 ABOUT HERE

1.2 Action slips and closure

Why do item and menu slips occur? As mentioned, they can occur because the release of the mouse button can move the mouse itself. However, both errors are also *action slips* (Reason, 1990). Action slips occur with expert users who perform many simple operations (such as button clicks and menu selections) automatically and do not explicitly monitor the feedback from each interaction. Lee (1992: 73) describes such errors thus: ‘as a skill develops, performance shifts from ‘closed loop’ control to ‘open-loop’ control, or from monitored mode to an automatic, unmonitored mode of processing’. As users become familiar with a simple task they no longer monitor the feedback from it so closely. Reason (1990: 8) suggests two conditions must be met for one of these errors to occur: ‘The performance of some largely automatic task in familiar surroundings and a marked degree of attentional capture by something other than the job in hand’. In this case the automatic task is the menu selection (which most users are very familiar with) within a familiar windowing system and the attentional capture comes from the main task that the user is performing.

One problem that exacerbates the difficulties of actions slips is *closure* (Dix *et al.*, 1993). Closure occurs when a user perceives a task as being completed. In some cases the task may appear to be completed when it is not. The user may experience closure and carry on to do something else and cause an error. Dix and Brewster (1994) and Brewster *et al.* (1995a) investigated such a situation that could occur with on-screen graphical buttons. In this case, users could slip off a graphical button by mistake and not press it. This happened because as the graphical button highlighted (when the mouse button was pressed), users experienced closure and carried on to their next task. In fact the graphical button was not actually pressed until the mouse button was released. After analysis, Brewster *et al.* suggested three criteria necessary to make action slips and closure a problem. In terms of menus these would be:

- i) The user reaches closure after the cursor is moved into the desired menu item and the text is highlighted.
- ii) The visual focus of the next action is at some distance from the target menu item.
- iii) The cursor is required at the new focus.

In this case, when the desired menu item has been highlighted (mouse moved over an item with the mouse button pressed down) the user experiences closure. The task is, in reality, not completed until the mouse button is released over the desired menu item. However, because of closure the user may begin to move on to his/her next task before

the menu selection has been completed. If this task requires the cursor and visual focus to be away from the menu, all the conditions for an action slip have been met.

An example of this error consider selecting 'Save' from the File menu in a word processor. The user will be more concerned with the document being saved than the mechanics of the menu, making action slips likely. He/she would be typing in the document and decide to save. The mouse (and visual attention) would be moved up to the menu to select Save, the menu item would be highlighted, the user would see the highlight, reach closure and then begin to move back to the document to continue typing. If this return movement is done too soon, the mouse may slip on to another item or out of the menu before the interaction is complete. The graphical feedback indicating this (either the highlight of the menu item the user slipped on to, or the removal of the highlight) is not demanding enough to grab the user's attention, therefore the error will not be noticed. In addition the menu only remains on the screen for a very short time after it has been used so the user has no chance to look back at it to see if everything is correct. In many systems no feedback is given to indicate a save has been performed, so the user would have no other indication of his/her error.

1.3 Overcoming the problems

In order to solve the problems of action slips and accidental mouse movements with menus users must perceive the feedback from the menu. Reason (p 157) says 'Making a postslip attentional check does not of itself ensure the detection of the error. Detection must also depend upon the availability of cues signalling the departure of action from current intention'. Therefore the right feedback must be given to ensure users know what is going on. The Macintosh-style multiple flash goes some way to dealing with this problem, but is not a complete solution. The highlighting of the menu item the user slipped on to happens for such a short period of time that it is very hard to see. In this paper we suggest using auditory feedback to solve the problems. Why use sound, why not just use extra graphical feedback? It is difficult (although not impossible) to solve these problems with extra graphics. Graphics displayed on the menu will not be seen by users because their attention will have moved back to the main task they are trying to perform (point ii above). The visual system has a narrow area of focus which means that users cannot look at the menu as well as their main task (Brewster, 1994). Information could be displayed at the mouse location indicating the result of the menu selection but we do not know if users are looking there either. In this case, sound has certain advantages. It can be heard from all around, it is good at getting our attention whilst we are looking at something else and it does not disrupt our visual attention. If we give menu selection information in sound then we

do not need to know where users are looking. If users must look at the menu then it forces them to stop what they are doing in their main task and causes the interface to intrude upon that task, sound does not have these drawbacks.

1.4 Previous work in the area

As mentioned above Brewster *et al.* (1995) investigated the use of sound to improve interaction with graphical buttons. Buttons have a subset of the problems of menus because users can slip off a button in a similar to a menu-slip. To solve the problems sounds were added when the mouse button was pressed down in a graphical button and when a correct selection was made. The results of adding sound to buttons significantly improved their usability. Error recovery was significantly faster and required fewer mouse clicks than standard graphical buttons. Participants also significantly preferred the sonically-enhanced buttons. The knowledge gained from that experiment suggested that we might be able to overcome some of the problems of menus in the same way.

Other work on sonifying widgets, for example scrollbars (Brewster *et al.* 1994, Beaudouin-Lafon and Conversy, 1996), have proved successful in improving usability. For example, Brewster *et al.*'s sonically-enhanced scrollbar significantly reduced both the time taken to complete tasks and the overall workload. We therefore believed that sound would allow us to improve the usability of menus.

Much work has been done on the optimal layout and design for menu structures (see Norman for a full review) but little has been done to investigate the selection of items from menus. A notable exception is the research by Walker *et al.* (1991) who proposed some design improvements for graphical menus to increase their usability by decreasing the time taken to move the cursor to the desired item. Their two main approaches were: To investigate the use of impermeable borders around menus and to increase the size of items further away from the start point in the menu to make them easier to hit. It was found that the use of impermeable borders (ones that the mouse could not move through) reduced the time taken to make a selection. The user could move the mouse to the top, bottom or right edge of the menu but not be able to move through. Therefore, he/she could aim to overshoot the menu item at the edge where there was an impermeable border knowing that it was not possible to do so. This eliminated the time taken to make the secondary movement required to correct the initial movement which overshoot or undershot the target item. The user could leave the menu without making a selection by moving the mouse out the left edge of the menu. One problem with this solution was that it did make the menus less flexible because users had to leave them from the left side only.

Walker *et al.*'s other solution was to increase the point size of menu items as their distance from the top of the menu increased, i.e. increasing the size of the target area. This allowed the user to make his/her movement towards the target more quickly. It did, however, also make the distance to the target greater. On large menus, the length of the menu was increased greatly. If the menu became too long, two movements would have to be made to reach the target, thus eliminating any benefits gained. Using sound to solve the problems would avoid these drawbacks.

There has been some previous work on adding sound to menus, although to solve different problems. Karshmer *et al.* (1994) added sounds to a menu system to aid navigation for blind users. They were not trying to overcome the slip off problems discussed here but to aid users in navigating around menus structures. The navigation was implemented by either changing the tones and timbres of the items in the menus, or by using synthetic speech to tell users their position in the hierarchy. Each menu had a different timbre, and as the user navigated up and down the menu, the pitch of the sounds increased and decreased accordingly. When an item was clicked it would say its name. Other work in the area of menu navigation in sound has been done by Barfield *et al.* (1991) and Brewster *et al.* (1996). Navigation was not the focus of our work here, our aim was to overcome slip off problems in graphical menus for sighted users.

Gaver (1986, 1989, 1992, 1993) did significant early work on the use of non-speech sounds in human-computer interfaces. He developed the idea of *Auditory Icons* and used them to sonify the Macintosh Finder interface to produce the *SonicFinder* (Gaver, 1989). Auditory icons are based around natural, everyday sounds that have an intuitive meaning in the interface. The SonicFinder added sounds for selecting different types of items, dragging and copying (Gaver did not add sounds to the menus in the SonicFinder). For example, different types of objects were given different base sounds: applications had a metallic sound, folders a paper sound and files a wooden sound. When selecting an application users would hear a deep metallic sound for a large application and a higher metallic sound for a small application. The extra sonic feedback gave users information that they could not see, for example file size and type when selecting. However, one thing that Gaver did not do was formally experimentally test his sonic enhancements to see if they actually improved usability. His results were more anecdotal. The addition of sounds to menus had to be fully tested to ensure they improved usability.

2. Annoyance

Whenever one talks about the use of non-speech sound at the human-computer interface there is always one question that is asked: Won't using sound be annoying? The main concern potential users of auditory interfaces have is annoyance due to sound pollution. This problem must be dealt with before there is general acceptance of sonically-enhanced interfaces.

Firstly, what is annoyance? It is defined by Berglund *et al.* (1994: 1501) 1 as: 'a feeling of displeasure associated with any agent or condition known or believed by an individual or a group to be adversely affecting them'. In this case the agent of interest is sound. There are two aspects to annoyance: A sound may be annoying to the user whose machine is making the noise (the primary user) and/or annoying to others in the same environment who overhear it (secondary users). Buxton (1989) has discussed some of the problems of sound and shown that there is no such thing as absolute silence; we all have to put up with varying amounts of noise in our working environments. He suggests that some sounds help us (information) and some impede us (noise). We therefore need to control sounds so that there are more informative ones and less noise. Sounds should be used in ways that provide useful information to the user. If they do this effectively then they will be informative; the user will want the information they contain. If the sounds are just gimmicks then they will be turned off by the user as they are just noise. Of course, one person's informative sounds are another's noise. It is therefore important to make sure that the sounds on one computer are not annoying for a colleague working nearby.

2.1 Previous studies of annoyance

There are few studies that particularly look at the problems of annoyance due to non-speech sounds in computers. One reason for this is that the use of sound in computers is still in its infancy. There are, however, many studies of annoyance from the sounds of aircraft, traffic or other environmental noise (for example, Broadbent 1972, Kuwano *et al.* 1988). Most of these suggest that the primary reason for the annoyance of sound is excessive intensity (Swift *et al.* 1989, Berglund *et al.* 1990). Even though these studies are not on sounds used in computers (which tend to be much quieter than the ones tested in research on annoyance from environmental sounds and may be used in much quieter office circumstances), excessive intensity is still likely to be the main cause of annoyance. There is also much work on the annoyance due to speech sounds, for example Berglund *et al.* (1994). Again, there are many problems due to the intensity of sounds used.

In a different context, Patterson (1982, 1989) investigated some of the problems with auditory warnings in aircraft cockpits. Many of the warnings were added in a 'better safe than sorry' manner which lead to them being so loud that the pilot's first response was to try and turn them off rather than deal with the problem being indicated. This is a similar approach to that taken by computer designers who add very loud error beeps to make sure that they are heard by the primary user, whether or not they are sitting right in front of the computer in a quiet office. Other techniques from the research into warning design (Stanton, 1994) can help in the design of sounds used for computers, as will be described below.

2.2 Annoyance from non-speech sound in computers

A loud sound grabs the attention of the primary user, even when the sound is communicating an unimportant event. As the sound is loud, it travels from one machine to the ears of other people working nearby, increasing the noise in their environment. In a normal office environment, computer users sit close to their workstations so that quiet sounds can easily be heard. Think, for example, of the noise of a hard disk that can be used as a check on activity within a system. Sound output from the computer should be at a level just above the threshold of background noise. The sounds would then be audible but not intrusive. There are problems with the sound output hardware of many computers that make this difficult:

- The loudspeaker is often of poor quality and mounted deep within the case of the machine (often near the fan). This means the sounds cannot be turned down to a low level without distortion.
- Loudspeaker problems can also lead to difficulties of localisation (Moore, 1997): Users can find it difficult to tell where a sound came from; they cannot tell if it came from their machine or a colleague's because of distortion due to bad positioning of the speaker. To overcome this the volume must be increased.

If sounds from the computer are loud then this can make two of the advantages of sound, discussed above, into potential reasons for annoyance. The first potential problem is that sound output is public whereas graphical output is private. This is advantageous because you can hear what your computer is doing, even if you are not looking at the screen. This can be useful, for example, when waiting for a long file copy to finish. The user does not have to be sitting looking at the computer, he/she can be working on something else and hear when the copy is finished. However, this can cause annoyance to colleagues working nearby. Just as the primary user can hear when the file

copy is finished, secondary users may also hear it. For the primary user the sound is information, but for secondary users the sound may be noise.

The second (related) problem is that sound is attention grabbing. This is an advantage in that sound can communicate important information but it can be a disadvantage in that all sounds may be communicated in an attention-grabbing way, even when not necessary. The attention-grabbing nature of sounds is useful because it means that alarms and warnings can be effectively indicated to users of a system. This is linked to the public nature of sound - the sounds can be heard even if the user is not looking at the screen of the computer. Again, this can lead to two different issues of annoyance. The primary user does not want all sounds in his/her environment to be communicated in an attention-grabbing way, some things are not important enough. For example, having an attention-grabbing sound indicating the arrival of email may be annoying if email arrives frequently. The second issue is similar to the problems of sound being public above. If warnings are presented in a demanding way then they can be overheard by others users nearby, again potentially causing annoyance.

2.3 How can annoyance be avoided?

Better quality audio hardware can help overcome some of the problems of annoyance. This can be achieved by manufacturers adding better quality loudspeakers to their machines. This is now becoming more common with 'multimedia' computers that have either external speakers or speakers mounted at the edges of the monitor. These provide better sound quality and less distortion which means that sounds can be turned down to a lower level without reducing quality. Stereo speakers also provide improved localisation of the sound source so that users can tell if the sound is coming from their machine.

There are also other techniques that can be used to overcome annoyance, even on poorer quality hardware. If sounds are kept to a low level then they are less likely to annoy other computer-users in the vicinity as they will not be able to hear them. Sounds giving constant feedback should be kept to just above the threshold of background sound. Sounds to provide notification of important events should be loud enough to capture the user's attention but still quiet so that others are not forced to hear them. This is possible because, as mentioned above, most people sit very close to their machines.

Headphones could be used so that sounds are only heard by the primary user of the system. This is not a perfect solution as many users want to be able to hear the other sounds in their environment (for example, the telephone)

and communicate with colleagues. Conventional headphones also physically tie users to their computers so that it is hard to move around.

Sound designers often want some of their sounds to capture the listener's attention. This is most often done by playing a loud sound. This is crude but effective. However, as it is potentially annoying for the primary user and those nearby, other methods are recommended. Listeners are not good at making absolute intensity judgements (Buxton *et al.*, 1991). Therefore, intensity should not be used on its own for differentiating sounds. If it must be used in this way then there would have to be large differences between the intensities used. This may lead to annoyance.

Manipulating sound parameters other than intensity can make sounds attention grabbing (but not annoying). Rhythm or pitch can be used to make sounds attention-grabbing (perhaps combined with a small intensity increase) because the human auditory system is very good at detecting dynamic stimuli. If a new sound is played, even at a low intensity, it is likely to grab a listener's attention (but not that of a colleague nearby). As another example, if the rhythm of a sound is changed (perhaps speeding up or slowing down) this will also demand attention. Edworthy *et al.* suggest other techniques for making sounds attention-grabbing without the use of intensity: High pitch, a wide pitch range, rapid onset and offset times, irregular harmonics and atonal or arrhythmic sounds (for more see Edworthy *et al.* (1989, 1991). The opposites of most of these can be used to make sounds avoidable.

Overall control of volume is normally given to the user by a volume knob so that he/she can adjust it as necessary. The volume of an auditory display is similar to the brightness on a graphical display. On a monitor the user can change the brightness of the display in response to the ambient light level. If the room is light then the brightness of the display will be increased so that the information on the screen can still be seen. If the room is dark then the brightness will be turned down so that the screen does not hurt the eyes. The volume control on a monitor acts in a similar way. If the room is noisy then the loudness will be increased to avoid masking by other sounds. If the room is quiet then the loudness will be reduced to avoid irritation. If the sounds used at the interface vary widely in loudness then turning up the volume so that the quiet sounds can be heard will cause the loud sounds to become irritating. Conversely, turning down the loud sounds to a pleasant level may cause the quiet ones to fall below the level of background sound. Therefore, keeping the sounds used within a narrow intensity range will avoid some of the problems of loudness due to annoyance.

The sounds used in the experiment described in this paper were designed with the above guidance in mind to try and reduce any problems of annoyance that might occur.

3. Experiment

As there had been no other work to try and overcome menu selection errors with sound we decided to run an experiment to discover if auditory feedback could improve usability. The aim was to enhance standard graphical menus with more salient feedback to see if menu errors could be solved and also to see if sound was effective as the feedback. For the experiment we needed to be able to differentiate four types of errors:

- a) an item slip,
- b) a menu slip,
- c) a slip onto a divider or disabled item,
- d) a mis-selection.

We could easily differentiate b) and c) because we could tell where the mouse was released (on a menu or not, on a divider or not). In principle, we could differentiate d) because we knew where the user was supposed to be clicking. We would be running a controlled experiment and the order of menu clicks would be pre-defined by the tasks we set, therefore if the wrong menu item was selected we would know. However, it was difficult to differentiate a) from d) because how could we tell if users slipped off one menu item on to another or were really choosing the one on which they finally released the mouse? How could we tell if they made a mis-selection (and just chose the wrong menu item) or slipped on to an item they did not want?

We conducted an initial investigation into menu interactions to study ways in which we might be able to find out if an item slip had occurred. We noticed that when users selected an item correctly they stayed on the item longer than when an item slip occurred. We therefore decided to use timing information to help us identify correct selections and item slips. In a pre-test we added a timer to a simple menu system so that we could investigate how long it took users to select items. From this we discovered that for a correct selection users held the mouse over the item for an average of 265 msec. before releasing. However, for an item slip the cursor was over the item for just 117 msec. For correct selections users moved the mouse and did some cognitive processing to ensure it was the correct menu item before releasing the mouse button. For an item slip there was no cognitive processing, just the mouse button release time. This gave us a method for identifying correct selections and item slips.

This also gave us a way of differentiating item slips from mis-selections. By using the timing information we could tell if the user had slipped on to an incorrect item or chosen it on purpose - a mis-selection would take the same amount of time as a correct selection and because we were running a controlled experiment we knew where the user was supposed to be clicking. We therefore had the information needed to be able to identify the errors associated with menu usage.

3.1 Experimental hypotheses

The main hypotheses were:

- The extra salient feedback for menu errors should make the task easier than with standard menus because users will be able to tell they have made errors and recover from them more readily. Sound (as a particular instance of such salient feedback) should be an effective way of providing this extra feedback when vision is already overloaded. This should result in an overall reduction in subjective workload and in particular a reduction in the effort required for the task.
- There should be no increase in annoyance due to the sounds as they will be providing information that the participants need to perform the task.
- The time taken to recover from item and menu slips should be reduced because users will know that they have made errors more quickly with the extra salient feedback. The number of errors corrected by the participants (i.e. corrected before an error dialogue comes up to say that an error has been made, meaning that the participant noticed the error) should also be increased as the demanding auditory feedback will be noticed.

3.2 Main task

Figure 2 shows the interface to the task the participants had to perform during the experiment. It was based on a car parts ordering system. Participants had to send parts to destinations using a menu system. In the figure the Parts and Destinations menus are shown. At the bottom of the screen the boxes with 'fan belt' and 'Auchterarder' show participants the part and destination to be chosen. Once a part and destination pair had been chosen, the participants selected 'Accept' on the File menu. If one or both of the chosen items was incorrect an error dialogue was displayed and corrections had to be made before participants could continue. If an item or menu slip occurred on any of the

Parts, Destinations or File menus then the participant would not be able to carry on to the next pair until he/she had noticed and corrected the error. The task was designed to be simple so that the participants could easily learn it and reach a level of automaticity in the task where slip-off errors would occur. Participants worked alone.

FIGURE 2 ABOUT HERE

The menus in both conditions provided all of the standard graphical feedback given by the Apple Macintosh. When a menu was chosen it dropped down displaying the list of items. As the user moved the mouse over the items they highlighted in reverse video, when an item was chosen it flashed three times, if the user moved off the menu (or released on a greyed-out item or a divider) there was no flash, the menu was just removed. If an item slip occurred the item slipped on to was highlighted and flashed before the menu was removed.

3.3 Sounds

Sounds were added to standard graphical menus as a method of providing the necessary salient feedback to overcome the problems described above. The sounds used were based around structured audio messages called *Earcons* (Blattner *et al.* 1989, Brewster *et al.* 1993). Earcons are abstract, musical tones that can be used in structured combinations to create sound messages to represent parts of an interface. The sounds were created using the earcon guidelines proposed by Brewster *et al.* (1995b). The sounds were all generated on an Apple Macintosh via MIDI using a Yamaha TG100 sound synthesiser and presented to participants via headphones. Previous work has shown that, if designed carefully, earcons are not annoying (Brewster *et al.*, 1995a).

Three earcons were needed to deal with the interaction problems described above:

1. An earcon was played when a menu was displayed. We used a family of organ sounds to indicate that the menus were related (part of the same application). The organ timbre was used as it has a wide pitch range and a slow attack (the initial part of the sound). Slow attack is one of the attributes that makes a sound less annoying. The File menu had a percussive organ, the Parts menu a drawbar organ and the Destinations menu a rock organ (General MIDI numbers 18, 17, 19). A low intensity, continuous note at pitch C₃ (523 Hz) was played for each of the menus. The sound continued as long as the cursor was in the menu. If the user moved the cursor out of the menu the sound stopped. This lack of feedback indicated to the participants that a menu slip had occurred. The method proved effective in the design

of sonically-enhanced buttons (Brewster *et al.*, 1995a) so was used again here. The sound was kept to a low intensity so that it would not be annoying.

2. To deal with the problem of item slips a combination of two earcons was used. A highlight sound was created that was similar to the standard graphical highlight. This was again a continuous, low intensity tone which used the timbre of the menu that the cursor was in. The sound alternated in pitch from B₂ (987 Hz) for odd numbered items and E₃ (329 Hz) for even numbered items. For example, the first item in the menu would be played at E₃, the second at B₂ and the third at E₃. This sound started when the mouse had been over an item for half a second. Only two different sounds were needed to indicate the movement from one menu item to another. These pitches were chosen to make the two earcons sound distinct. This sound was stopped if the user moved the mouse out of the menu (menu slip) or moved over a divider or disabled item.
3. The final earcon indicated a selection. This could be either a correct selection or an item slip. For a correct selection (the mouse was in a menu item for longer than 117 msec.) the earcon played was based around the timbre of the menu the cursor was in and the pitch of the sound was based on the highlight sound for the menu item. Using these as a base, two 40 msec. duration tones were played at a higher intensity. To indicate an incorrect selection the timbre of the current menu was again used. However, this time a fixed rhythm of three notes of 40 msec. each at pitch C₂, B₂ then F₂ was played (these sounded discordant and therefore attention grabbing without using intensity). This was not dependent on the highlight sound for the current menu item. This sound was always the same, indicating the item slip error in each of the different menus. If the user released the mouse over a divider then no sound was played.

To avoid potential annoyance due to the sounds we made sure they were all played at low volume. The menu earcon (1) indicated a menu slip by a lack of sound and the highlight sound (2) did not start playing until the participant had been over an item for half a second. In a normal, fast interaction this sound was not played, the user only heard the menu sound (1) and the selection sound (3). We tried to minimise the number of sounds so the primary user of the computer would not be annoyed and also to minimise the volume so that others nearby would not hear the sounds (Berglund *et al.*, 1990).

3.4 Participants

Twelve participants were used. They were undergraduate and postgraduate students from the Department of Computing Science at the University of Glasgow. Three were female, nine male. All had more than three years experience of graphical interfaces and menus. Expert participants were used because the type of error studied here was an action slip (see above).

3.5 Experimental design and procedure

The experiment was a two-condition, repeated-measures within-groups design (see Table 1). The order of presentation was counterbalanced to avoid any learning effects across conditions. One group performed the auditory menu condition first and the other used standard visual menus first. Training was given before each condition. During each condition the participants had to enter 150 parts to destinations pairs as quickly as possible. Instructions were read from a prepared script.

TABLE 1 ABOUT HERE

To get a full measurement of usability we used a range of quantitative and qualitative measures (Bevan and Macleod, 1994). These were total time taken, error rates and subjective workload (Hart and Staveland, 1988). Time and error rate reductions would show quantitative improvements and workload reductions would show qualitative improvements.

Hart and Staveland (1988) break workload into six different factors: mental demand, physical demand, time pressure, effort expended, performance level achieved and frustration experienced. NASA has developed a measurement tool, the NASA-Task Load Index (TLX) for estimating these subjective factors. We used this but added a seventh factor: Annoyance. As mentioned, this is one of the main concerns of users. In the experiment described here annoyance due to auditory feedback was measured to find out if it was indeed a problem. We also asked our participants to indicate overall preference, i.e. which of the two interfaces they felt made the task easiest. Participants had to fill in workload charts after both conditions of the experiment.

4. Results

4.1 Workload results

Figure 3 shows the average workload score for each category. They were scored in the range 0-20. The average workload (based on the six standard workload factors: mental, physical, time, effort, frustration and performance) was 10.02 (Standard Deviation (SD) = 2.5) for the auditory condition and 11.04 (SD = 2.5) for the visual. This difference was significant ($T_{11}=4.42$, $p=0.007$) indicating that the participants found the task easier with the sonically-enhanced menus. This confirmed the hypothesis.

FIGURE 3 ABOUT HERE

Paired *T*-tests were carried out on the auditory versus visual conditions for each of the workload categories. Average effort was significantly reduced from 13.9 (SD = 3.7) in the visual condition to 12.5 (SD = 3.1) in the auditory condition ($T_{11}=2.43$, $p=0.03$) confirming the hypothesis. All the other scores showed no significant differences.

There was no significant difference in terms of annoyance ($T_{11}=1.43$, $p=0.18$). Six of the participants rated the visual condition more annoying, five the auditory condition more annoying and one rated them the same (auditory: average = 8.75, SD = 5.1, visual: average = 10.5, SD = 6.3). This indicated that the participants did not find the auditory feedback to be annoying confirming the hypothesis.

4.2 Time and error results

The overall number of errors of both types corrected (i.e. before the error dialogue appeared) increased significantly from an average of 55% (SD = 33.5) to 92% (SD = 5.5) in the auditory condition ($T_{11}=3.982$, $p=0.002$), confirming the hypothesis. The average percentage of item slips corrected increased significantly to 70% (SD = 12.1) in the auditory condition from 17% (SD = 36.5) in the visual condition ($T_{11}=4.21$, $p=0.001$), again confirming the hypothesis. The percentage of menu slips corrected increased to 100% in the auditory condition (all of the menu slips made were corrected) from 63% (SD = 26.3) in the visual.

The average time taken to recover (detection and correction of the error) from all errors decreased significantly in the auditory condition (2.38 secs., SD = 0.73) compared to the visual condition (4.18 secs., SD = 1.8) ($T_{11}=4.04$, $p=0.0009$). Figure 4 shows the average time each participant took to recover from an error in each condition (both

item and menu slip errors are included). The average time taken to recover from an item slip decreased significantly from 4.1 secs. (SD = 3.4) in the visual condition to 2.53 secs. (SD = 2.2) in the auditory condition ($T_{11}=2.55$, $p=0.03$). The recovery time from a menu slip decreased from an average of 3.07 secs. (SD = 2.9) in the visual condition to 1.57 secs. (SD = 0.1) in the auditory condition. A wide variance in the times in the visual condition meant that an *F*-Test was conducted. This showed a significant reduction in the variance in the auditory condition ($F_{11}=18.11$, $p=1.76 \times 10^{-5}$). The range of times decreased from a minimum of 1.87 secs and a maximum of 11.25 secs. in the visual condition to between 1.73 secs. and 2.08 secs.

There was no learning effect over the conditions. Comparing the time taken to complete tasks in the first condition versus the second showed no significant effect ($T_{11}=0.3$, $p=0.769$).

FIGURE 4 ABOUT HERE

4.3 Discussion

The results have shown that by indicating menu and item slip errors in a salient way the usability of menus can be significantly improved. Sound was shown to be a very effective method of providing this feedback. The workload analysis showed that the overall workload, and in particular effort expended, was reduced significantly when using the sonically-enhanced menus. Workload was reduced because the sonic enhancements meant that participants needed to expend less effort to notice and recover from menu and item slips. This, however, was not at the expense of making the menus more annoying to use. This, along with previous results from sonifying other graphical widgets (Brewster *et al.* 1994, Brewster *et al.* 1995a, Brewster 1998), indicates that earcons can provide a significant qualitative improvement in a user's experience with a system.

There was no difference in terms of annoyance. This indicated that the sounds were not annoying to the main user of the computer. However, the sounds could be annoying for others working nearby. In our experiment each participant was wearing headphones and working alone, in real use of such a sonically-enhanced interface the user may not use headphones and, if he/she works in a shared office, there is the possibility of others nearby overhearing the sounds and for them to be annoyed. As mentioned, better sound hardware can overcome many of these problems. Using the guidance on sound design described above can further reduce the problems as the main cause of annoyance, intensity, need not be used to make the sounds attention grabbing.

The results also showed that the integration of sound allowed faster recovery from errors. Both menu and item slip error recovery times showed a significant improvement in the auditory condition. Participants could tell they had made an error and correct it significantly faster than in the visual condition. The reason was that the users' visual focus moved to the next menu before the previous selection was completed. In the visual condition the avoidable graphical menu feedback was missed, whereas in the auditory condition the demanding sonic feedback ensured the user was alerted to any problems.

Participants spent significantly less time correcting errors in the auditory condition (on average 25.99 secs.) than in the visual condition (44.72 secs.) due to the faster recovery times. However, the overall time taken for the task was not significantly reduced in the auditory condition ($T_{11}=0.73$, $p=0.48$). This was because the users took a non-significantly longer time to make a correct selection in the auditory condition (on average 2.14 secs. compared to 2.10 secs.). As each user made approximately 452 correct selections during each condition, this total came to an extra 18.08 secs. (452×0.04). This compensated for the 18.73 secs. saved correcting errors. The 0.04 sec. increase in the time taken to make a correct selection may have been caused by the time it took the participants to hear the auditory feedback or by delays in the operating system when sending data to the MIDI synthesiser for playback. One way to overcome this might be to reduce the length of the sounds used. However, the difference was so small that in normal interactions the benefits gained from correcting errors more reliably would offset this very small increase in time.

The item slip sound was sometimes played incorrectly. This could happen if a participant interacted with the system quickly and their correct selections took less than 117 msec. These false positives could occur because some users moved the mouse around much faster than others. To solve this problem, a future system could include a control panel, similar to the panel that allows users to set their preferred double-click speed, which would allow setting of preferred menu selection speed.

5. Future work

As mentioned, the sounds used here are only one possible way that the necessary salient feedback on the menu errors could be presented, there will be a whole range of other solutions. Further work is needed to investigate these others. For example, could better graphics be used to give the extra feedback? Another interesting area would be the

investigation of haptic feedback to indicate the errors. This would again have the advantage of putting no extra burden on the visual sense.

This work is part of an overall project to develop a sonically-enhanced interface toolkit (Brewster, 1995). The toolkit will consist of all of the standard widgets but they will be enhanced with sound to improve their usability.

One interesting area of future work would be to investigate the use of automatic error correction. Our pre-test showed that we could identify item slips by the time they took. This means that we could add an auto correction feature to the menus so that if a selection took place that the system identified as an item slip then it could be rejected and the adjacent item where the mouse had been for a longer amount of time could be selected. This would avoid errors of this type occurring. We will investigate the use of sound together with automatic error correction in a future experiment.

The use of non-speech audio at the human-computer interface is still in its infancy. As yet, there have been no long-term studies of the problems of annoyance due to sound. In the experiment described above, participants only listened to sounds for 30 - 45 minutes. For this time period the sounds may have still had some novelty value that the participants liked. A long term study, over some weeks or months, would show if the sounds became annoying after long-term exposure. It is hoped that they would not annoy because they provide information that improves performance, and are not just noise.

As mentioned, the overall sound volume is under the control of the user. If he/she turns up the volume too much it may affect others working nearby. There is little an auditory interface designer can do about this, it is in the hands of the user. One potential future solution to this problem would be to have some form of automatic volume control. The computer would monitor the surrounding sound level and increase or decrease the volume of the its own sounds so that they would be just above this background level. Of course there would have to be a strictly-controlled maximum limit to which the system could raise the volume otherwise feedback could occur: in a room of computers fitted with such devices the volume would be raised by one then another in turn to make its own sounds audible. If a system such as this was carefully designed then it would prevent one user turning up his/her sounds and annoying others nearby.

6. Conclusions

Standard graphical menus have problems because they do not provide salient feedback to the user for menu and item slips. The experiment described in this paper has shown that, if these errors are indicated, usability can be improved. The experiment also showed that non-speech sounds can present this extra feedback very effectively.

This work has demonstrated the power of multimodal interactions. By combining the advantages of the visual and auditory senses problems when interacting with menus can be overcome. The results have shown that integrating earcons into graphical menus can reduce the subjective workload required to use them. It can also increase the number of errors corrected and reduce the time taken to recover from errors. These advantages can be gained without making the menus more annoying to use.

Using sound in this way means that users can concentrate on their main task without the interface intruding. The combination of graphics and sound provided a significant qualitative and quantitative improvement in the participants' experience with the menus. Future interface designers can use sound in this way and know that they will be improving the usability of their systems.

7. Acknowledgement

Part of this work was funded by EPSRC Grant GR/L79212.

8. References

References by Brewster plus sound samples are available from <http://www.dcs.gla.ac.uk/~stephen/>

Alty, J. L., 1995, Can we use music in human-computer interaction?, In M.A.R. Kirby, A.J. Dix and J.E. Finlay (eds), *Proceedings of HCI'95*, (Cambridge University Press), pp. 409-423.

Barfield, W., Rosenberg, C. and Levasseur, G., 1991, The use of icons, earcons and commands in the design of an online hierarchical menu, *IEEE Transactions on Professional Communication*, **34**, 101-108.

Beaudouin-Lafon, M. and Conversy, S., 1996, Auditory illusions for audio feedback, In M. Tauber (ed.), *ACM CHI'96 Conference Companion*, (ACM Press, Addison-Wesley), pp. 299-300.

- Berglund, B., Harder, K. and Preis, A., 1994, Annoyance perception of sound and information extraction, *Journal of the Acoustical Society of America*, **95**, 1501-1509.
- Berglund, B., Preis, A. and Rankin, K., 1990, Relationship between loudness and annoyance for ten community sounds, *Environment International*, **16**, 523-531.
- Bevan, N. and Macleod, M., 1994, Usability measurement in context, *International Journal of Man-Machine Studies*, **13**, 123-145.
- Blattner, M., Sumikawa, D. and Greenberg, R., 1989, Earcons and icons: Their structure and common design principles, *Human Computer Interaction*, **4**, 11-44.
- Brewster, S. A., 1994, Providing a structured method for integrating non-speech audio into human-computer interfaces, PhD Thesis, University of York, UK.
- Brewster, S. A., 1995, The development of a sonically-enhanced widget set, In B. Blumenthal, J. Gornostaev and C. Unger (eds), *Proceedings of EWHCI'95*, (International Centre for Scientific and Technical Information), pp. 126-129.
- Brewster, S. A., 1998, Using earcons to improve the usability of tool palettes, In C.-M. Karat and A. Lund (eds), *Summary proceedings of ACM CHI'98*, (ACM Press, Addison-Wesley), pp. 297-298.
- Brewster, S. A., Raty, V.-P. and Kortekangas, A., 1996, Earcons as a method of providing navigational cues in a menu hierarchy, In A. Sasse, R. Cunningham and R. Winder (eds), *Proceedings of BCS HCI'96*, (Springer), pp. 169-183.
- Brewster, S. A., Wright, P. C., Dix, A. J. and Edwards, A. D. N., 1995a, The sonic enhancement of graphical buttons, In K. Nordby, P. Helmersen, D. Gilmore and S. Arnesen (eds), *Proceedings of IFIP Interact'95*, (Chapman & Hall), pp. 43-48.
- Brewster, S. A., Wright, P. C. and Edwards, A. D. N., 1993, An evaluation of earcons for use in auditory human-computer interfaces, In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (eds), *Proceedings of ACM/IFIP INTERCHI'93*, (ACM Press, Addison-Wesley), pp. 222-227.

- Brewster, S. A., Wright, P. C. and Edwards, A. D. N., 1994, The design and evaluation of an auditory-enhanced scrollbar, In B. Adelson, S. Dumais and J. Olson (eds), *Proceedings of ACM CHI'94*, (ACM Press, Addison-Wesley), pp. 173-179.
- Brewster, S. A., Wright, P. C. and Edwards, A. D. N., 1995b, Experimentally derived guidelines for the creation of earcons, In M. Kirby, A. Dix and J. Finlay (eds), *Adjunct Proceedings of BCS HCI'95*, pp. 155-159.
- Broadbent, D. E., 1972, Individual differences in annoyance by noise, *Sound*, **6**, 56-61.
- Buxton, W., 1989, Introduction to this special issue on nonspeech audio, *Human Computer Interaction*, **4**, 1-9.
- Buxton, W., Gaver, W. and Bly, S., 1991, Tutorial number 8: The use of non-speech audio at the interface, In S. Robertson, G. Olson and J. Olson (eds), *Proceedings of ACM CHI'91*, (ACM Press, Addison-Wesley).
- Crease, M., 1996, Making Menus Musical, Undergraduate Thesis, University of Glasgow.
- Dix, A., Finlay, J., Abowd, G. and Beale, R., 1993, *Human-Computer Interaction*, (Prentice-Hall: London).
- Dix, A. J. and Brewster, S. A., 1994, Causing trouble with buttons, In *Ancillary Proceedings of BCS HCI'94*.
- Edworthy, J., Loxley, S. and Dennis, I., 1991, Improving auditory warning design: Relationships between warning sound parameters and perceived urgency, *Human Factors*, **33**, 205-231.
- Edworthy, J., Loxley, S., Geelhoed, E. and Dennis, I., 1989, The perceived urgency of auditory warnings, *Proceedings of the Institute of Acoustics*, **11**, 73-80.
- Gaver, W., 1986, Auditory Icons: Using sound in computer interfaces, *Human Computer Interaction*, **2**, 167-177.
- Gaver, W., 1989, The SonicFinder: An interface that uses auditory icons, *Human Computer Interaction*, **4**, 67-94.
- Gaver, W., 1992, Using and creating auditory icons, In G. Kramer (ed.), *Proceedings of ICAD'92*, (Addison-Wesley), pp. 417-446.
- Gaver, W., 1993, Synthesizing auditory icons, In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (eds), *Proceedings of INTERCHI'93*, (ACM Press, Addison-Wesley), pp. 228-235.

Hall, L. and Bescos, X., 1995, Menu-what menu?, *Interacting with Computers*, **7**, 383-394.

Hart, S. and Staveland, L., 1988 Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. Hancock and N. Meshkati (eds), *Human mental workload*. (North Holland B.V.: Amsterdam), pp. 139-183.

Karshmer, A., Brawner, P. and Reisswig, G., 1994, An experimental sound-based hierarchical menu navigation system for visually handicapped use of graphical user interfaces, In E. Glinert (ed.), *Proceedings of ACM ASSETS'94*, (ACM Press), pp. 123-128.

Kuwano, S., Namba, S. and Fastl, H., 1988, On the judgement of loudness, noisiness and annoyance with actual and artificial noises, *Journal of Sound and Vibration*, **127**, 457-465.

Lee, W. O., 1992, The effects of skill development and feedback on action slips, In A. Monk, D. Diaper and M.D. Harrison (eds), *Proceedings of HCI'92*, (Cambridge University Press), pp. 73-86.

Moore, B. C., 1997, *An Introduction to the Psychology of Hearing*, 4th ed. (Academic Press: London).

Norman, K., 1991, *The psychology of menu selection*, (Ablex Publishing: Norwood, NJ, USA).

Patterson, R. D., 1982, Guidelines for auditory warning systems on civil aircraft, CAA Paper No. 82017, Civil Aviation Authority, London.

Patterson, R. D., 1989, Guidelines for the design of auditory warning sounds, *Proceeding of the Institute of Acoustics, Spring Conference*, **11**, 17-24.

Reason, J., 1990, *Human Error*, (Cambridge University Press: Cambridge, UK).

Stanton, N., (ed.) 1994, *Human factors in alarm design*, (Taylor & Francis Ltd: London, UK)

Swift, C. G., Flindell, I. H. and Rice, C. G., 1989, Annoyance and impulsivity judgements of environmental noises, *Proceedings of the Institute of Acoustics*, **11**, 551-559.

Walker, N., Smelcer, J. and Nilsen, E., 1991, Optimizing speed and accuracy of menu selection: A comparison of walking and pull-down menus, *International Journal of Human-Computer Studies*, **35**, 871-890.

Participants	Condition 1	Condition 2
Six Participants	Auditory Menus Train & Test	Visual Menus Train & Test
Six Participants	Visual Menus Train & Test	Auditory Menus Train & Test

Table 1: Format of the experiment.

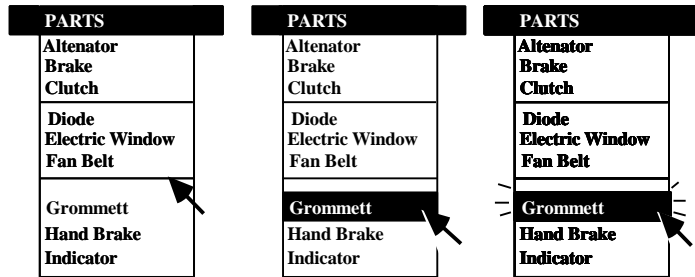
Table 1: Format of the experiment.

Figure 1: Examples of (i) a correct selection, (ii) an item slip, (iii) a menu slip.

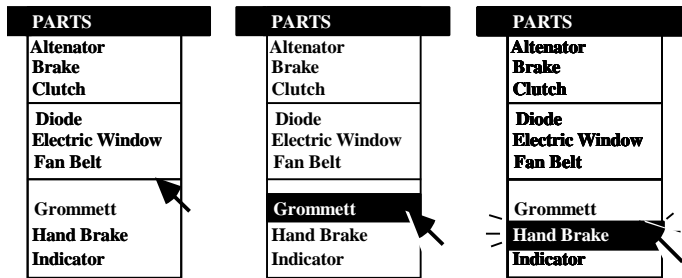
Figure 2: The screen of the menu testing program. The parts and destinations menus are shown.

Figure 3: Average workload scores for the two conditions. In the first six categories higher scores mean higher workload. The final two categories, performance and overall preference, are separated because higher scores mean less workload.

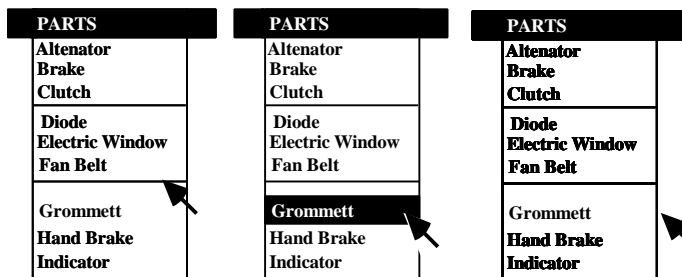
Figure 4: Average error recovery times for all errors.



(i) A correct menu selection. The cursor is moved into grommet and released. Grommet flashes before the menu disappears.



(ii) An Item Slip. The cursor is moved into grommet and as the button is released, the cursor slips into Hand Brake. Hand Brake flashes before the menu disappears.



(iii) A Menu Slip. The cursor is moved into grommet and as the button is released, the cursor slips off the side of the menu. The highlight disappears. The same happens if the cursor slips into a divider or disabled item.

Figure 1: Examples of (i) a correct selection, (ii) an item slip, (iii) a menu slip.

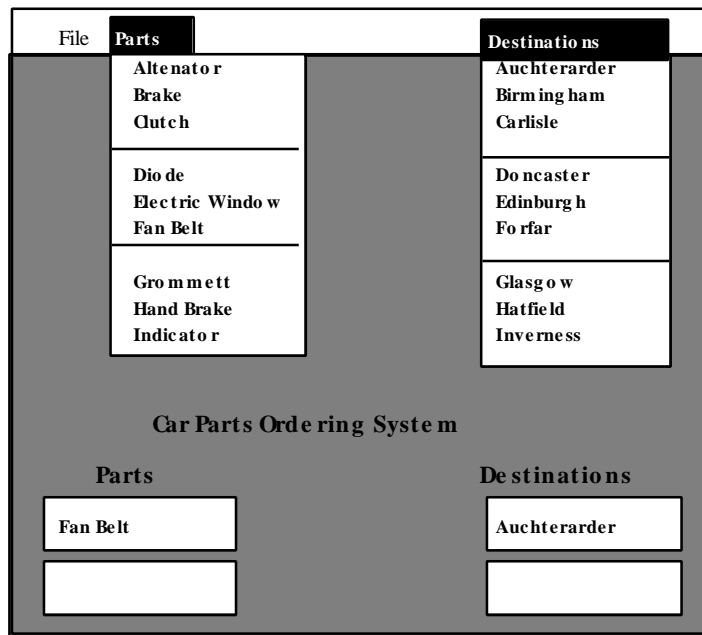


Figure 2: The screen of the menu testing program. The parts and destinations menus are shown.

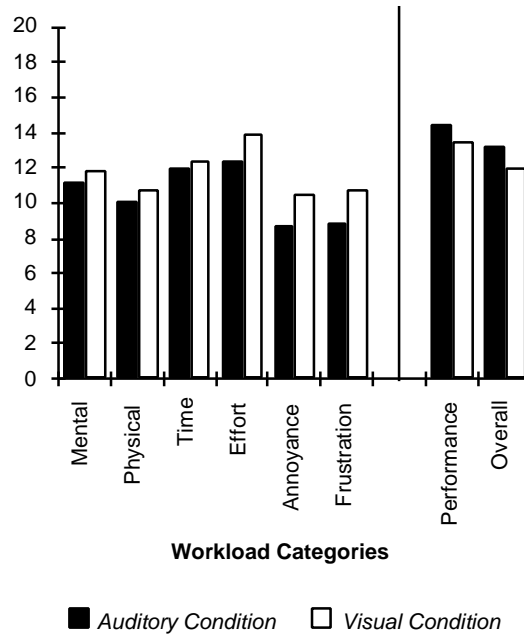


Figure 3: Average workload scores for the two conditions. In the first six categories higher scores mean higher workload. The final two categories, performance and overall preference, are separated because higher scores mean less workload.

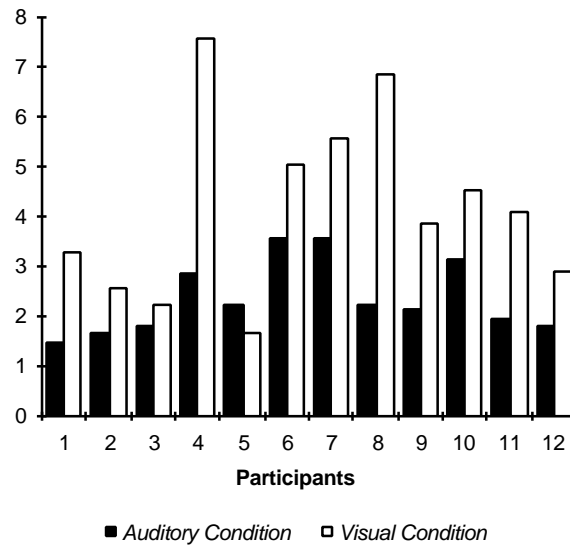


Figure 4: Average error recovery times for all errors.