Sonically-Enhanced Drag and Drop

Stephen A Brewster

Department of Computing Science, University of Glasgow, Glasgow, G12 8RZ, UK
Tel: (0)141 330 4966, Fax: (0)141 330 4913
Email: stephen@dcs.gla.ac.uk, Web: www.dcs.gla.ac.uk/~stephen/

Abstract

This paper describes an experiment to investigate if the addition of non-speech sounds to the drag and drop operation would increase usability. There are several problems with drag and drop that can result in the user not dropping a source icon over the target correctly. These occur because the source can visually obscure the target making it hard to see if the target is highlighted. Structured non-speech sounds called *earcons* were added to indicate when the source was over the target, when it had been dropped on the target and when it had not. Results from the experiment showed that subjective workload was significantly reduced, and overall preference significantly increased, without sonically-enhanced drag and drop being more annoying to use. Results also showed that time taken to do drag and drop was significantly reduced. Therefore, sonic-enhancement can significantly improve the usability of drag and drop.

1 Introduction

'Drag and drop' is a feature common to most graphical human-computer interfaces; it appears in applications from graphics packages to word-processors. It allows the user to select an object in the interface, drag it and then drop it over something else (see Figure 1A and B). For example, the user might select a text file icon and then drag it to and drop it on the icon for a text editor to edit it. In many systems the target highlights when the mouse is over it to indicate that an object may be dropped (Figure 1B). This is a very natural way of interacting (and is in fact one of the foundations of the direct manipulation style if interaction) and provides the user with great flexibility; he/she can drag the text file to different applications, disks, printers or the wastebasket as required.

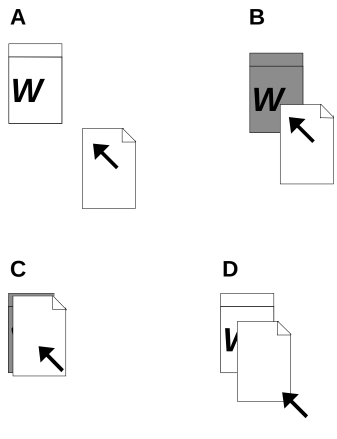

Figure 1: Examples of drag and drop operation.

There are, however, several problems with drag and drop. It can be difficult to hit a target and drop things on it (especially if the target is small). Gaver [11-13] noted this problem in the Macintosh Finder ([12], p 81):

> "A common problem in hitting such targets comes when the object, but not the cursor, is positioned over the target. In this situation, dropping the object does not place it inside the target, but instead positions it so that it obscures the target further."

He calls the problem 'chasing the trashcan'. Figure 1C shows the situation where the source icon is almost obscuring the destination one, so that it is hard to see if the destination is highlighted or not. In this example the target is highlighted showing that the icon can be dropped. Figure 1D shows the situation Gaver describes: the mouse pointer is not over the target (so the target is not highlighted) but the source icon is almost obscuring it, making it very difficult for users to see if the target will accept the icon or not.

It is difficult to know if the icon being dragged can be dropped on a particular target. For example, a graphics file cannot be dropped on a text editor. The target icon may not highlight when the mouse is over it indicating that it cannot open the source icon. This lack of feedback can be confusing because it is exactly the same as that given when the mouse is not over the target. In some systems the target will highlight but nothing will happen if the object is of a type the target cannot deal with, again this is confusing.

Sounds can potentially overcome the problems of drag and drop. Gaver [12] attempted to solve it in his *SonicFinder* - a version of the Macintosh Finder which used non-speech sounds (p 81):

> "Auditory confirmation that a target has been hit turns out to be one of the most obviously useful features of the SonicFinder, especially in finding small folder icons that may be partially obscured by overlapping windows. …The auditory cue indicates a true hit, and so reduces the amount of time spent playing 'chase the trashcan' ".

Gaver used a clinking sound in his SonicFinder to indicate when an item was over a target application. A similar approach was used here. When a source icon was dragged over a target that would accept it a quiet background sound was played. This indicated to users that the icon could be dropped and also cued them when to drop it (whether the target was obscured or not, users could still hear the sound). The sound was low-intensity (to avoid annoyance) and was played for as long as the mouse was over the target.

Although there has been little other work on the use of sound in drag and drop, there has been some on the use of sound to improve other graphical widgets. Brewster and colleagues have successfully improved the usability of buttons, scrollbars, tool palettes and menus with sound [4-6, 9]. They reduced the time taken to recover from errors, to complete tasks, and workload without any increase in annoyance. Beaudouin-Lafon & Conversy [1] added sound to solve usability problems in scrollbars. They used an auditory illusion called Shepard-Risset tones which appear to increase (or decrease) in pitch indefinitely. When the user was scrolling down a continuously decreasing tone was used, when scrolling up an increasing one. If scrolling errors occurred then the user would hear tones moving in the wrong direction. Results from these studies suggested that sound would be effective in solving the problems with drag and drop.

2 Experiment

Gaver [12] did not formally test to see if his enhancements in the SonicFinder improved usability. An experiment was therefore needed to see if the addition of sound to drag and drop would solve the usability problems described above. The experiment was a counterbalanced, two-condition, within-groups design. Each participant performed both the visual condition (standard drag and drop) and auditory condition (sonically-enhanced drag and drop). Table 1 shows the format of the experiment.

<table>
<tr><th></th><th>Condition 1</th><th></th><th>Condition 2</th><th></th></tr>
<tr><td>Nine Participants</td><td>Sonically-enhanced drag and drop
train and test</td><td rowspan="2">Workload test</td><td>Visual drag and drop
train and test</td><td rowspan="2">Workload test</td></tr>
<tr><td>Nine Participants</td><td>Visual drag and drop
train and test</td><td>Sonically-enhanced drag and drop
train and test</td></tr>
</table>

Table 1: Format of the experiment.

To get a full measurement of usability quantitative measures of time and error rates were taken, along with qualitative measures of subjective workload (NASA TLX), annoyance and user preference [2]. Training was given before each condition and workload was assessed after each condition. Each condition lasted approximately 20 minutes. Instructions were read from a prepared script.

2.1 Participants

Eighteen participants were used. They were all computing science students from the University of Glasgow with more than three years of experience with graphical interfaces and drag and drop.

2.2 Hypotheses

The main hypotheses were that: The sounds should reduce the overall workload of drag and drop as the more salient feedback will make the task easier. This should be demonstrated by lower overall workload and lower effort when sound was used. There should be no increase in annoyance due to the sounds as they will be providing information that the participants need to overcome usability problems and are, therefore, not just gimmicks. The total time taken to perform drag and drop operations should be reduced as the more salient feedback will indicate to users when they are over a target more effectively, allowing them to work faster.

2.3 Sounds Used

The sounds used were based around structured non–speech musical messages called *Earcons* [3, 8]. Earcons are abstract, musical sounds that can be used in structured combinations to create audio messages to represent parts of an interface. The earcons were created using the earcon guidelines proposed by Brewster *et al.* [10]. The sounds were played on a General MIDI synthesiser controlled by an Apple Macintosh and presented to participants by loudspeakers. A full Java demonstration of the sounds and the sonically-enhanced drag and drop widget can be found at http://www.dcs.gla.ac.uk/~stephen/

Three earcons were used. As suggested by the earcon guidelines, the different sounds needed were assigned different timbres. Each timbre was chosen to be distinctive and different from the others so that users would not confuse them.

The first earcon was played when the participant dragged the source icon over a target. This was a quiet, continuous reed organ sound at pitch C_4 (130Hz). The sound started when the source was dragged over the target and stopped when it was moved off. This sound helped the user correctly position the source icon over the destination - the sound could always be heard even if the destination was visually obscured by the source (see sound example 1 for a demonstration of this sound).

The second earcon was used to indicate that the user had successfully dropped the source on to the target. The sound was at pitch C_4 and was played for 300 msec. with a tinkle bell timbre (similar to Gaver's SonicFinder sound. Sound example 2 demonstrates this sound. You hear the first earcon as the user moves over the target then the bell sound as the user drops the source onto the target). If the user was expecting to drop a source icon on to a destination and this sound was not heard it would indicate to them that they had missed the target (this method was successful with sonically-enhanced buttons [5] so was used again here).

The third earcon was played when the user released the mouse but did not drop the source on the target (i.e. the target was missed or the use decided not to drop on it). The sound was again at pitch C_4 and was played for 300 msec. with an orchestral hit timbre. This timbre was attention grabbing (see sound example 3 for three examples of this sound). Sound example 4 gives four examples of complete interactions with the sonically-enhanced drag and drop widget: the first two show successful interactions; the third example demonstrates the user missing the target completely; the final example shows the user moving over the target then moving off and dropping the source.

These three sounds together helped to overcome the problems described above. The first earcon would help users to position over the target correctly before dropping (whether the target was visually obscured or not), the second would indicate that a successful drop had been made and the third would indicate that the target had been missed.

2.4 Experimental Tasks

Participants had to perform two tasks during each condition of the experiment. Task 1 involved drag and drop in a visually complex interface, the second was drag and drop in a simple interface. Figure 2 shows a screenshot of the complex interface. For this task, users had to drop the lettered file icons into the appropriate target folders, as fast as possible. Once this had been done the screen was cleared and replaced with the icons in a different order (this order was the same for each condition and each participant). This was done five times.

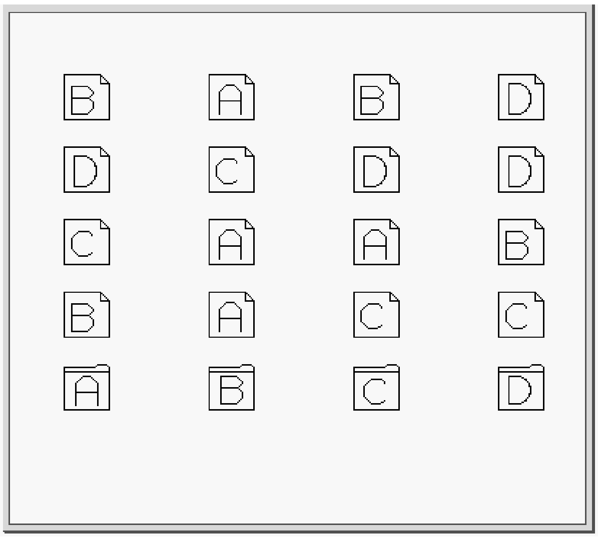

Figure 2: Screenshot of the first experimental task.

For the second task in each condition a simple interface was used with one file and folder (see Figure 3). In this case the file icon always appeared in the same place at the bottom of the screen but, after each successful drop, the screen was cleared and the target folder was moved to a new random location and the user had to drag the file to that. The set of random locations was the same for each condition and each participant. Each participant had to do this one hundred times (giving a total of 160 drags and drops for both tasks in each condition).

The drag and drop widget used the standard visual cues described above: the target highlighted when the user's mouse pointer moved over it when dragging the source icon, the highlight was removed if the user moved the mouse pointer off the target. The highlight was also removed when the source was dropped on the target.

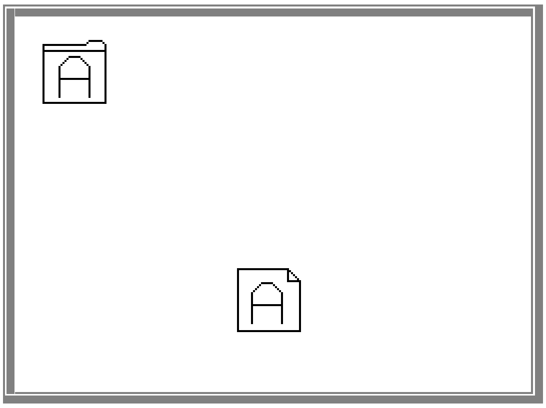

Figure 3: Screenshot of the second experimental task.

3 Qualitative Results

The average NASA TLX workload scores for each category are shown in Figure 4. They were scored in the range 0–20. There was a significant reduction in overall workload in the auditory condition (measured using the standard six factors: T_5=4.03, p=0.01). Overall workload fell from 8.24 in visual condition to 7.51 in the auditory. There was a significant reduction in effort expended and a significant increase in overall preference for the auditory condition (effort: T_{17}=2.94, p=0.009, overall preference: T_{17}=2.64, p=0.017). This confirmed the hypotheses. There was no effect due to annoyance (T_{17}=0.94, p=0.36) with nine participants rating the auditory condition more annoying, six rating the visual condition more annoying and three rating them the same. This confirmed the hypothesis.

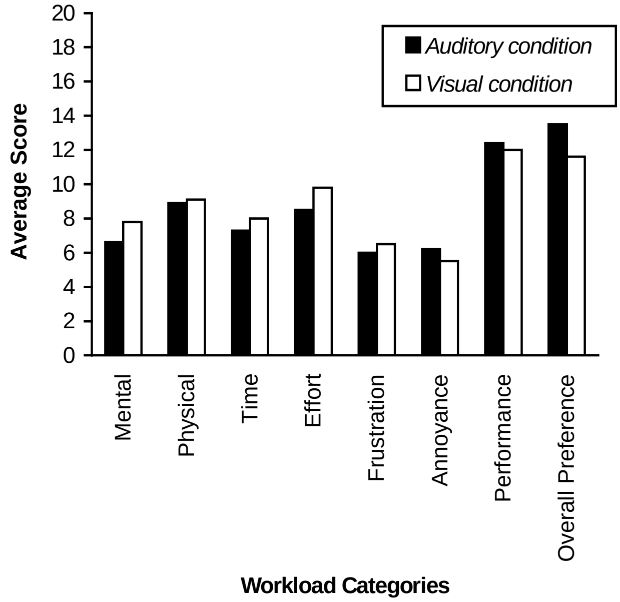

Figure 4: Average workload scores for the two conditions. In the first six categories higher scores mean higher workload. In the final two categories higher scores mean lower workload.

4 Quantitative Results

The total time taken to complete both tasks was significantly reduced in the auditory condition (Task 1: T_{17}=3.09, p=0.006, Task 2: T_{17}=3.79, p=0.001). For example, the average time to complete Task 2 was 107.2 sec. in the auditory condition and 114.0 sec. in the visual. This confirmed the hypothesis.

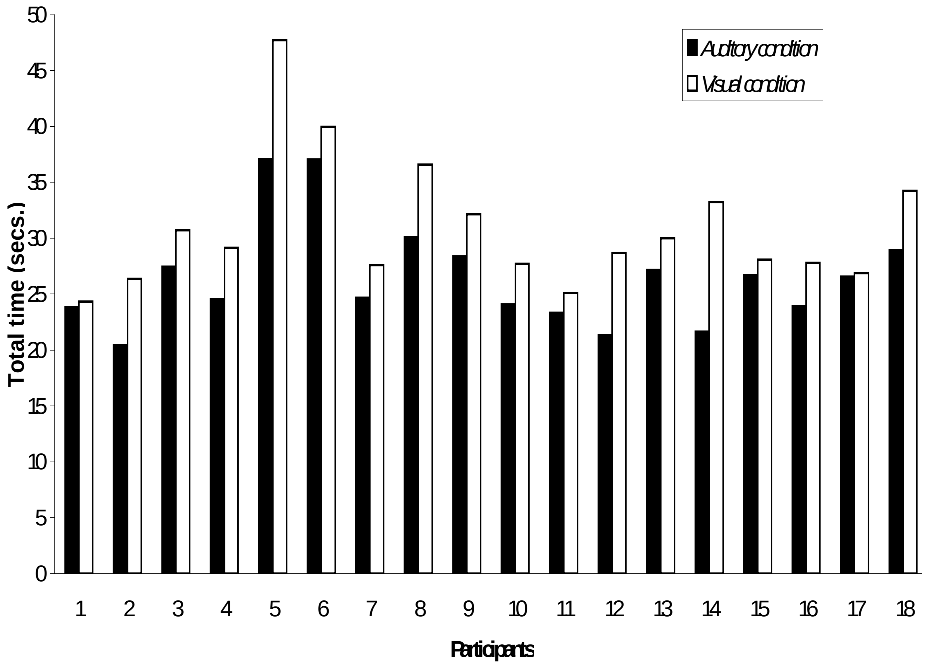

Figure 5: Target highlight time for Task 1.

The target highlight time was measured for each drag and drop. This was the length of time that the target icon was highlighted before the source icon was dropped on it (i.e. when the mouse was over the target and the mouse button was being held down). Figure 5 shows the results for Task 1. In both tasks in the auditory condition the highlight time was significantly reduced (Task 1: T17=6.00, $p=0.000014$, Task 2: T17=6.71, $p=0.0000036$). In the auditory condition the average total highlight time for both tasks was 59.4 sec., in the visual condition the time was 70.3 sec. Therefore, in the auditory condition participants saved close to 11 seconds over both tasks. This accounted for the significant difference in the total time taken in each task.

5 Discussion and Conclusions

The workload results confirmed the hypotheses that the addition of earcons significantly reduces the subjective workload associated with drag and drop and, in particular, the effort required. The sound also increased overall preference but did not make drag and drop more annoying for the users. This result adds further weight to previous research that has shown substantial qualitative improvements from the addition of sound [4, 7, 9].

The earcons significantly reduced the time taken to perform drag and drop. The detailed analysis showed that this was due to the reduction in time spent with the source icon held over the target - the audio highlight was much more effective than the visual at indicating to users when they were on target, therefore they could drop the source icon more quickly. The main reason for this was that the audio highlight could not be obscured, as could the graphical. Because drag and drop is such a common operation across many different types of applications speeding it up will give substantial benefits.

This research shows that the earcons overcame the problems of ‘chasing the trashcan’ where the source icon visually obscures the target. Presenting the information in sound avoided this problem, as users did not need to be able to see the target visually highlight to know they were over it – they could hear it even if the target was totally obscured. This demonstrated that, in this case, sonic highlighting was much more effective than visual and showed that the addition of simple sounds to an interface can significantly increase usability.

6 Acknowledgements

Many thanks go to Alasdair Horne for his work on this research as part of his fourth year undergraduate project in the Department of Computing Science, University of Glasgow. This research was part funded by EPSRC Grant GR/L79212.

7 References

All references by Brewster can be found at http://www.dcs.gla.ac.uk/~stephen/

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

2. Bevan, N. and Macleod, M. Usability measurement in context. *International Journal of Man-Machine Studies* 13, 1 & 2 (1994), 123-145.

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

4. Brewster, S.A. Using Non-Speech Sound to Overcome Information Overload. *Displays* 17 (1997), 179-189.

5. Brewster, S.A. The design of sonically-enhanced widgets. *Accepted for publication in Interacting with Computers* (1998).

6. Brewster, S.A. Using earcons to improve the usability of tool palettes. In *Summary proceedings of ACM CHI'98* (Los Angeles, Ca) ACM Press, Addison-Wesley, 1998, pp. 297-298.

7. Brewster, S.A. and Crease, M.G. Making Menus Musical. In *Proceedings of IFIP Interact'97* (Sydney, Australia) Chapman & Hall, 1997, pp. 389-396.

8. Brewster, S.A., Wright, P.C. and Edwards, A.D.N. An evaluation of earcons for use in auditory human-computer interfaces. In *Proceedings of ACM/IFIP INTERCHI'93* (Amsterdam, Holland) ACM Press, Addison-Wesley, 1993, pp. 222-227.

9. Brewster, S.A., Wright, P.C. and Edwards, A.D.N. The design and evaluation of an auditory-enhanced scrollbar. In *Proceedings of ACM CHI'94* (Boston, MA) ACM Press, Addison-Wesley, 1994, pp. 173-179.

10. Brewster, S.A., Wright, P.C. and Edwards, A.D.N. Experimentally derived guidelines for the creation of earcons. In *Adjunct Proceedings of BCS HCI'95* (Huddersfield, UK), 1995, pp. 155-159.

11. Gaver, W. Auditory Icons: Using sound in computer interfaces. *Human Computer Interaction* 2, 2 (1986), 167-177.

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

13. Gaver, W. Synthesizing auditory icons. In *Proceedings of INTERCHI'93* (Amsterdam) ACM Press, Addison-Wesley, 1993, pp. 228-235.