

An Audio User Interface

for A Music and Sound Editor.

by Julio d'Escriván, original concept and research

and Jan Trutzschler, programmer and developer

An auditory user interface for a music and sound editor (PAL202)

Table of contents

Report on project PAL 202	2
Sample Materials	7
Videos	8
The TeaTracks manual	8
Getting started with the TeaTracks multitracker.	8
1. <i>getting around</i>	8
2. <i>Adding a file to the soundpool and then to a track</i>	8
3. <i>Getting Information and roaming around</i>	9
5. <i>locating to a specific time on the timeline</i>	10
6. <i>Bounce audio</i>	11
7. <i>Save your Project</i>	11
<i>Summary of keyboard shortcuts:</i>	11
Bibliography including relevant further references on auditory displays	12

Report on project PAL 202

The auditory user interface for a music and sound editor arose out of the need to look after one of my own visually impaired students at Anglia Ruskin University. The original project is rather ambitious as it aims to provide, eventually, a software application of professional standard. Thanks to the TechDis/HEAT scheme, we have been able to create a working prototype which can already be used by students, although more money and time are needed to fully realise this idea and make it even more accessible and reliable. Our brief was to create a simple software application that would allow a visually impaired user to aurally access the controls in a sound editing environment.

Software music and sound editing environments allow users to select audio files which will then be placed in different time positions relative to each other, to cut, copy and paste from and into them, and finally to mix the audio down to a single stereo file (this process is called bouncing). These environments are also referred to as 'audio sequencers' or simply 'sequencers'. The problem is that their user interfaces are prominently based on visual paradigms, usually with few or no accessibility options. The challenge for us was to use custom software programming to embed these options from within the sequencer itself.

In July we started work by creating a specification of what we thought the software should be able to do. The following are my notes to Jan Trutzschler, who carried out the programming of the auditory feedback in SuperCollider, an 'Object Oriented Programming' language. Jan had been working on a visual multitracking application and his experience with sound sequencing was a key element in the realisation of this project.

"I suggest we go for sfx audio feedback because the machine talking is so obtrusive as you try and navigate the interface, as we have found from using the mac voice over. For example we could have short sounds that signify each action, a sort of audio equivalent to visual icons. These sounds could be either a sample or a synthetic sound.

When the user first begins to utilise the auditory user interface multitracker, they should receive both the spoken feedback followed by the sfx. I can design these sounds myself. The idea is that once the user can relate the word 'play' to, for example, a short click such as you would get from pressing a button, then he can shut off the speaking part and just have the click sound. This would make their work very speedy and efficient as they don't have to wait for the computer to pronounce the whole word.

At any given point the user should be able to query what a certain command is by using a qwerty key modifier, for example, that would speak out the command they are trying to use -this is in case they have forgotten what a sound stands for. I did this on the FM synth because I found that once the student knew what keys did, he quickly began to change the sound and now and again he could confirm what he was doing by pressing the 'query' key.

*Every command must be actioned by a key or combination of keys, or by an assignable controller number, we can negotiate these, but I believe from teaching experience that transport commands should live on the numeric keypad as this makes them very quick to use.. e.g., enter->play, '0'-> return to zero, '-' ->forward, "-"->backward, *->record, "."->locate ... (and of course spacebar should also be 'play' as in all sequencers !)"*

As expressed in these notes, the need to provide sounds or auditory icons (Gaver, 1986) was obvious from the start. A consideration of issues involved in communicating graphical information using musical elements (Alty & Rigas, 1998) was also interesting but the demands of getting a basic application working made us concentrate on speech and short sfx as audio feedback. What we did not envisage at that stage, but we hope to incorporate into future versions, were *earcons* (Blattner & Sumikawa, 1989) and *spearcons* (Walker, Nance & Lindsay, 2006). Earcons are abstract melodic elements that can be associated with user interface elements. Spearcons are rapid speech based sounds that have been shown to enhance accessibility in auditory displays (this is what auditory user interfaces are generally called in the human-computer interface research field). What we found as the project developed is that since Jan had to program 'from the ground' up in order to include these accessibility features into his previously existing multitracking object oriented classes, we had to make choices on what should be implemented in order to have a working version by the deadline of this project. This does not

mean that the ideas expressed above will not be implemented, simply that pragmatism has guided our choices during the creation process in order to meet the deadline.

I then proceeded to provide, in table form, a list of specific functionalities that the software should have regarding transport control and Jan suggested which combination of keys (keycommands) could be used to control them:

Command	spoken feedback	audio equivalent	confirmation sound or speech	keycommand
play (toggle, so we don't need a 'pause' command or a stop !)	'play'	short 'button press' sound	upon query, respond: 'playing'	spacebar
return to zero	'at zero'	two short blips	respond with now line position value	shift-ctrl-arrow <-
return to last played start position	'and again'	short 'button press' sound	upon query, respond: 'playing'	shift-arrow <-
rewind	'rewind'	(sound of analog tape rewinding: short sample) tick (short Synth)	'rewind'	arrow <-
forward	'forward'	sound of analog tape forwarding: short sample zwip (short Synth)	'forward'	arrow ->
scrub	"scrub"	the sound of scrubbing itself ! - assuming we can make this scrub, of course...	same as audio equivalent	
select location for the 'now' line, the computer waits for a set of numbers to be typed in, e.g. 3.1.1 to go to the 1st beat of the 3rd bar...	'locate'	a 'zip' sound implying movement	'locate'	(currently we work in absolute time) . then type: mm ss mm (minutes:seconds: milliseconds)
where am I? gives back a bar number or a timing, this key will be much used !!	responds with the spoken position in bars/beats/ticks or time in mm:ss:ff	the speech sound 'huh?'	responds with the spoken position in bars/beats/ticks or time in mm:ss:ff	for the cursor position: i for the currently selected object position: ctrl-i
record	'record'	a short feedback sample like you get from opening a mic for example, very short !	'recording'	
focus selection tabbing	current selection	toggle sound	you selected -- selection	ctrl-` (keep ctrl pressed and tab through with `)

I assumed these basic controls should be represented aurally in these way, through experimenting with the interface we eventually identified which ones worked best and added others. The next step was to find out how the manipulation of tracks and sound files could be made accessible, so time we tabulated more commands and I added some feedback upon testing.

Jan suggested that everything which has to do with selecting an audio region should use the option (alt) key, and everything, which actually moves audio regions should use the ctrl key.

Command	spoken feedback	audio equivalent	confirmation sound or speech	functionality already available in MultiTracker	keycommand (suggested by Jan)	jd'e comments
select a track	the track number	same	'track N'	yes		
select an object on track			speech			see below
select next object on track	"on track N selected" - soundfile name		speech		option-arrow right (->)	this should be renamed to 'move' the selected object
select previous object	"on track N selected" - soundfile name		speech		option- arrow left (<-)	same as above
add previous object to selection			speech		shift—option-arrow-left	this is actually to add another object to the group I want to move...
add next object to selection			speech		shift—option-arrow-right	same as above
move a region forward region can be an object or multiple objects (thus a selection)	'moving forward'	a 'click-click-click' sound as the position changes over the resolution, for example if the timing grid is set to 1/4 beats, then each time a region is moved we will hear a click every beat that it is displaced forward, I suggest the first click of the bar should always be loudest	as audio equivalent	yes	ctrl-arrow (right)	this seems to happen with ctrl ->

move a region backward	'moving backward'	same as forward but with a reverse envelope	same as fwd.	yes	ctrl-arrow (left)	
move a region to the track above	'moving up'	up in pitch sliding sound 'sssuiit'	'moving up'	yes *(1)	ctrl - arrow up	works great once the region is selected, but how to select the region without doing anything to it? just selecting it?
move a region to the track below	'moving down'	down in pitch sliding sound 'sssiiiut'	'moving down'	yes *(1)	ctrl - arrow down	same as imm above
select first region on track number					alt - 1 .. 7	doesn't seem to work...
select region	the region number like this for example: 'R 15'	same	same	yes		implemented differently, see manual below
trim beginning	'fIN'	same 'click-click-click' as when moving a region, this indicates to the user how much they are trimming	'trimming in'	yes (buggy)		not implemented
trim ending	'fOUT'	same as for trim beginning but with reverse envelope	'trimming out'	yes		not implemented
split region (audiofile)				yes	ctrl-y	works great!
locate beginning of region				no		to be implemented
locate beginning of region				no		to be implemented
reverse region				no		to be implemented
focus on panning				yes *(2)		to be implemented
focus on volume				yes *(2)		to be implemented
loop region				yes (does not work for soundfiles yet)		to be implemented
stretch region				no		to be implemented
remove region	removed' - soundfile name and track		removed -- name -- from track -- track		del (backspace)	works ok

**(1) What a track really is needs to be defined. At the current implementation it has not really any functionality besides the visual placement.*

But it is already possible to set Group to a certain track (see MTrack). That way the can be connected to the JTMixer.

**(2) panning and volume can be done via an envelope or by a fixed value directly in the MultiTracker's Synth or via the JTMixer.*

Finally, in the spirit of my early notes, Jan added the speech feedback for soundfile managing, this is an area which should benefit from working with earcons and spearcons in the next stage of our research and development.

function	audio fx	speech	keycommand
select soundfile	soundfilename		arrow up/down
add soundfile to current track on current position	none	"added soundfile -- name -- to track -- number -- at time -- time"	ctrl-a
add soundfile to new position new track (add popup)	none	"on track N adding soundfile, please type time and track number"	ctrl-alt-a
add one SoundFile or a selection of Soundfiles to the pool using a Dialog	none	none (you need to turn the mac VO for this, see manual)	ctrl-o
add a directory with Soundfiles to the pool	none	"added directory..."	ctrl-shift-o
pre-listen selected file	the soundfile itself	none	p (plays as long as the key is pressed)
remove file from pool	none	"removed -- name -- from track -- track"	del (backspace)

Once all these features were working, we proceeded to test them ourselves for consistency. We found issues of stability that were solved gradually, but in the end it is clear that we need more development time. The main tester was a blind PhD student named David Hindmarch. He uses a commercial sequencer but was thrilled at the prospect of a purpose built sequencer with auditory display. He raised many important issues and gave us ideas for further development. Jan had a meeting with him in Birmingham and he reported back: (please note that the working name for the sequencer is 'teatracks')

"On the 16th of November 2007 I met David Hindmarch, a blind student under Jonty Harrison at the University of Birmingham, to show him the current state of development of the multi-tracker sound program. First of all he was very excited to hear about the project. He has been using Windows PCs until now, on which he has been working with SoundForge and Sonar. These two programs are scripted in order to give spoken feedback and to be controlled with keyboard commands. However they were not designed for visually disabled users and David showed me a few examples of graphical interfaces, which act very intuitive, when one can see them, but otherwise are not accessible at all. David was very happy about all spoken and sonic feedback, which one can get from the multi-tracker "tea-tracks". And since he has a lot of experience with computer programs of that kind, he had some very useful suggestions, which made me realise how much one is focused on the visual representation even when trying to forget about them. He sketched out a work flow, which was partly already possible and which I took as a guideline for further development. The workflow can be described as the following: One imports audio files into the program's sound-file pool, moves the time cursor to certain position, places a sound file on a selected track and uses copy-, cut- and paste-operations on the sound-file. Multiple sound files can be placed and mixed to another file on the hard-disk. At the meeting I also met Zlatko Baracskaï, who has started to develop

some small applications for David. Since he also uses SuperCollider, as I do, we decided that it would be of advantage for all of us to try to incorporate these applications as plug-ins into "tea-tracks". Jan Trutzschler."

I also made contact with David Hindmarch and exchanged ideas along the same lines, in fact, regarding our project versus existing commercial sequencers he said this in an email exchange on 18th November 2007:

"Sound forge is becoming increasingly clunky and unreliable, if blind users of Computers can start to use readily designed Apps built in SC [SuperCollider], it will begin to level the playing field in regard to Sighted musicians."

In fact, quite early on, another interesting comment regarding inclusion was communicated to me via email by Dr. Tony Stockman, a blind senior academic researcher into auditory displays of Queen Mary, University of London, in May 2007:

I am very excited about this project, it seems to me remarkable that, as far as I am aware, no one has embarked on it before, as you say in the abstract, current interfaces provide a major barrier to visually impaired students of electronic music, there is a real need for something of this kind.

What we have done and intend to carry on pursuing has great potential in terms of enabling greater accessibility for the visually impaired. At present this software enables a blind musician to do basic sound editing and in that way it has been successful, but the real benefit of having developed this project is the research and development opportunities it opens. There is a real need to create applications that provide auditory displays for visually impaired users, especially in music. It seems that many blind people tend to be especially sensitive to auditory feedback and this works to their advantage in the fields of music and audio. Blind users who are also musicians are especially advantaged in that they can differentiate a greater range of sound meanings. Anecdotally, they seem also able to identify pitch and timbre with particular precision. What is needed is the development of music and audio tools that are based on these natural advantages as opposed to 'remedy' tools which attempt to translate sighted interfaces for non-sighted users. It requires a paradigm shift. I am very interested in this and I believe that what we have done, as well as having a concrete output, provides a useful starting point.

The methodology employed, that of identifying and implementing appropriate audio feedback is not exclusive to one programming language. We used SuperCollider because it is specifically designed for music, but the design guiding principles are not subject to this choice. Even the programming that has been done is simply an instance of an approach which can be applied in many different ways: that of systematically creating an auditory display for music editing based on auditory icons and speech. The advantage, though, of programming from the 'ground level up' as it were is great in that accessible functionality can be implemented at every stage. It also means that it is easier to adapt the software to new findings. For example, now that we have a software construct (a 'class') that can produce a sound effect in response to an action, it is a trivial matter to substitute it by a musical one (earcon) or a rapid speech sample (spearcon). It must be noted that the programming of spoken feedback by Jan was particularly creative and that he had to develop a parallel 'speech server' to that of the Apple Macintosh computer. Previous attempts I had made in this area were unsuccessful as the voice over utility conflicts with SuperCollider speech unless a separate class is developed. It must also be noted that although the functionality to respond to the USB mixing interfaces provided by TechDis is already enabled (although without aural representation yet), we had to postpone the level mixing capabilities of the sequencer as a matter of priority. We needed to have a working application and, as mentioned earlier, more research and development time is needed to provide all the desired functionalities.

Sample Materials

Please find together with this report the actual application TeaTracks. It is designed for the Apple Macintosh computer under OSX. What follows is a manual describing different workflow approaches to accomplishing editing tasks through our auditory display. You can also download short videos of blind user David Hindmarch using the interface through the following links.

The videos are simply 'slices' of our meeting where I consulted with David about the accessibility of the system and we tested various features.

Videos

http://bitbongo.com/for_techdis/addingFiles.mov

http://bitbongo.com/for_techdis/Focusing.mov

http://bitbongo.com/for_techdis/Navigating.mov

http://bitbongo.com/for_techdis/Placing.mov

http://bitbongo.com/for_techdis/splittingRegions.mov

The TeaTracks manual

Getting started with the TeaTracks multitracker.

A brief and informal manual for the use of an auditory display on the TeaTracks multitracker, a music and sound editor.

Notice that if you are using OSX version Leopard you may need to turn some global keyboard shortcuts off, as they may be used by TeaTracks, such as the arrow keys plus control which are used to navigate the available screens.

1. getting around

Launch the application TeaTracks, you will be welcomed and then you will find yourself focused on an arrange window with two main areas of the interface, the soundfile pool where all the audio files are kept, and to which you may add more, and the timeline where you can position your soundfiles to play at different times and to play simultaneously. There are two more windows but we are still developing them so there is no need to navigate away from the default window where for the moment everything will happen (By the way, if you ever need to stop the computer speaking just press 'escape'). This default window has two areas upon which you can focus, the sound pool

Let us start by pressing 'control - tilde'.

TeaTracks tells you that you have selected 'soundfiles'.

If you keep the 'control' key pressed, after you release the tilde, it will tell you 'focused selections 'tab on' ' and it will tell you which area of the window is selected. If you keep the control key down and press the tilde again, it will focus on the other one, and so on. At any point you can stop the computer speaking by pressing the 'esc' key.

In any case you can also get to the soundfile pool by simply pressing the letter 'a', independently of where you are focused.

You should be in the soundfile pool now.

2. Adding a file to the soundpool and then to a track

The soundpool is where all the soundfiles are kept. The computer simply refers to that area as "soundfiles" but it is useful to call it the soundpool.

Let's add a file: press control and the letter 'f' at the same time.

The computer speaks "open sound file dialogue" and now you need to switch on the Mac Voice Over utility to navigate to the file you want to edit. While you are learning the key commands to the application, you may want to leave the mac voiceover on, at least it uses a different voice, or you can choose a very different one so that you can differentiate them easily.

The computer speaks "added file" and the name of the file. At this point you are now focused on the timeline so you can perform operations there as you will see below.

If you want, now you may turn off the Mac Voice Over utility and return to the voice in TeaTracks.

Now you have a new file in your soundfile pool, if you would like to hear it, just press and hold the letter P, when you release it the file will stop playing. If you add more than one, you can navigate the list using the arrows up and down.

Let's move the new file into the timeline.

Press 'control-a', and the soundfile will be added to track zero at whatever time location we happen to be, at this stage, since we have not played anything, it should be at zero.

The computer speaks: "...added..." and tells you where it has put it. Alternatively, if you have a sound selected in the sound-pool just press 'enter'.

3. Getting Information and roaming around

If you press the letter 'i' at any time, you will be told where the cursor is and which track is selected. Try it. The computer tells you that you are focused "on track zero, current time zero seconds, zero milliseconds"

If you press 'control-i', the computer will tell you which soundfile you have selected and where it is playing from. The computer tells you "time position of soundfile A B C is X minutes, X seconds , X milliseconds"

Let's rewind to the very start of the timeline. Press shift-ctrl-left arrow

The computer speaks: "...to zero", and you also hear a dry sound, this is the auditory icon that tells you that you have reached time zero.

Now, let's just roam around the timeline first, we have eight tracks available and our soundfile is at track zero.

Make sure you are focused on the timeline, for this you can press control-tilde as explained earlier. remember to keep the control key held down, you can then toggle between the two areas by pressing the tilde.

Now, If we use 'alt-upArrow' or 'alt-downArrow' we can navigate to different tracks. try it now. Notice how when you select each track, it plays a number of fast ticks telling you which one you are on. The number of ticks correspond to the track number. You need to move up and down a bit to identify well this auditory icon, zero ticks is track zero (your first track) so it will have no ticks but if you move down a track then you will hear where you are (we will improve this for the next version). It is worth noting that sighted users also do this as they need visual confirmation that they are on the right track, so they flick the cursor up and down to highlight tracks for this confirmation. Our ticks are the auditory highlight. For example, track 5 will let us hear 5 ticks when it is selected in this way. As we explore the different tracks, you will need to use the alt-leftArrow or alt-rightArrow to select the very first object that is found there, then it will be read out loud by the computer. If you want to move the object to a different track use 'control-up-arrow' or 'control-down-arrow' and the object will be moved and the ticking aural feedback will be given to you by teatracks.

If you want to move the cursor backwards or forwards in time you use the left and right arrow keys.

Press the right arrow key. You hear a little shuffling noise, one for each unit of time you advance. It is setup to advance in increments of 100 milliseconds. If you now press the left arrow key, it will rewind and a different sound will be heard, like a kind of backwards sweep. It confirms you are rewinding.

Now let's select the sound file we placed on track zero and move it in time. Before we move it, let's confirm where it is by pressing control - i, then lets move it and consult again with control - i to see what the new position is.

Navigate to track zero using 'alt - upArrow' or 'alt - downArrow', once you are there pick up the object by pressing 'alt-leftArrow' (if you left the track at time zero) or 'alt-rightArrow'. Now that you have an object selected, when you move it with 'control-leftArrow' or 'control-rightArrow you also will hear forwards and backwards click sounds that confirm the direction in which you are moving the soundfile on the track.

Hey, we haven't played the sequencer yet, so now just press the space bar and you should hear your soundfile. You stop it by pressing the spacebar again. Use your key commands to forward or rewind and play from where you want.

By the way if you want to remove the selected soundfile from the timeline, just press backspace while it is selected and it will be taken off the timeline.

Now try adding another sound file. Use the key commands described above and place it on your timeline. It doesn't matter if you place it on the same track as the previous soundfile, they will not play together then but you can just pick one of them up and move it to the next track. You have enough information now to be able to do that, so go ahead and try it.

4. Split, copying and pasting a soundfile

Splitting a selected soundfile

To split a soundfile, first press the letter 'a' to focus on the soundfile pool. choose an audio file (if you have none then load one as explained above).

Press 'control-a' to place the soundfile at the position where your cursor is, you are now focused on the timeline (locate it to where you want by using the arrows).

Playback your sound file by pressing spacebar, when you hear the point at which you would like to split, stop the multitracker by pressing spacebar again

Now press control-y and the file will be split in two parts, you can choose either and move it to a different track or further along the same track.

Copy and paste

To copy or cut a region of a soundfile for use somewhere else

Press the letter 'a' to focus on the soundfile pool. choose an audio file (if you have none then load one as explained above).

Press 'control-a' to place the soundfile at the position where your cursor is, you are now focused on the timeline (locate it to where you want by using the arrows).

Playback your soundfile by pressing spacebar, when you hear the point at which you would like the target region to commence, press the letter 'r'. The computer will let you know that you have chosen the start of the region. When you hear the point at which you would like to stop, press the letter 'r' again. The computer will let you know you have chosen the end of the region. Now you can stop the sequencer.

You can use 'control-c' to copy that region or 'control-x' to cut that region onto the clipboard. You can then use the transport commands to find the spot where you would like to paste the region on the timeline. At that point use 'control-v' to paste it. If you need to undo anything, you can press control-z. If at any time you wish to mute a track, just press the letter "m" while that track is selected.

For now we have no auditory feedback for trimming the fadein and fadeout of a soundfile, so on this prototype all we do is split files and put them at different times and/or tracks on our timeline. We hope to implement this feature soon.

5. locating to a specific time on the timeline

We can directly enter a time position to play from, on the timeline, by typing the "." character. When you type it, the computer is waiting for you to enter the time where you want to go in the following format: minutes minutes seconds seconds milliseconds milliseconds milliseconds. For example, if you want to locate to 6 seconds, you would type the following sequence: 0 0 0 6 0 0 0

If a soundfile is selected and you press control-i then you can move that soundfile to a specific place by immediately entering the time in the following format: minutes minutes seconds seconds milliseconds milliseconds milliseconds. For example, if

you want to go to locate 6 seconds, you would type the following sequence: 0 0 0 6 0 0 0. The computer will announce the new location for your soundfile.

6. Bounce audio

Your audio files can be mixed down automatically by pressing control - b. Changing the levels is yet not implemented, again we expect to do that soon.

7. Save your Project

Your project can be saved by pressing control - s.

If you want to open a project you saved previously you can use 'control-o'.

8. Summary of keyboard shortcuts:

Editing

control - c	copy region
control - v	paste region on selected track
control - x	cut region
control - z	undo
r	set region start / end point, create region

File Management

control - o	open project
control - s	save project

SoundFile Management

control - f	add soundfiles to soundfile pool
control - b	bounce audio to disk
control - i	speak information about selected object
arrow-up	go to previous soundfile in the pool list
arrow-down	go to next soundfile in the pool list
p	preview currently selected file in pool
enter	add selected soundfile to track
delete	remove object from soundfile pool
a	select soundfile pool in order to add a soundfile to current track at current time

Transport Control

spacebar	play or stop
control - '.'	type in start time of selected soundfile: m m:sec sec: milli milli milli
control - shift - left arrow	return to zero
i	speak current time, current track
','	type in current time: m m:sec sec: milli milli milli
arrow-left	move time cursor, increase current time
arrow-right	move time cursor, decrease current time

Timeline actions

control-arrow-left	move selected soundfile on timeline, increase start time
control-arrow-right	move selected soundfile on timeline, decrease start time
control-arrow-up	move selected soundfile on track, decrease id
control-arrow-down	move selected soundfile on track, increase id
delete	remove object
alt-arrow-up	select track and select object on that track at current time position
alt-arrow-down	select track and select object on that track at current time position
alt-arrow-left	select next object on track
alt-arrow-right	select previous object on track

Bibliography including relevant further references on auditory displays

- ALTY, J.L. and RIGAS, D.I., 1998. Communicating graphical information to blind users using music: the role of context. *Proceedings of the SIGCHI conference on Human factors in computing systems*, , pp. 574-581.
- BLATTNER, M.M., SUMIKAWA, D.A. and GREENBERG, R.M., 1989. Earcons and Icons: Their Structure and Common Design Principles. *Human-Computer Interaction*, **4**(1), pp. 11-44.
- BREWSTER, S., 2003. NONSPEECH AUDITORY OUTPUT. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, .
- BREWSTER, S.A., WRIGHT, P.C. and EDWARDS, A.D.N., 1995. Parallel earcons: reducing the length of audio messages. *International Journal of Human-Computer Studies*, **43**(22), pp. 153-175.
- BREWSTER, S.A., WRIGHT, P.C. and EDWARDS, A.D.N., 1993. An evaluation of earcons for use in auditory human-computer interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems*, , pp. 222-227.
- FERNSTROM, M. and BRAZIL, E., 2001. Sonic Browsing: an auditory tool for multimedia asset management. *Proc.Int.Conf.on Auditory Display (ICAD)*.
- GAVER, W.W., 1989. The SonicFinder: An Interface That Uses Auditory Icons. *Human-Computer Interaction*, **4**(1), pp. 67-94.
- GAVER, W.W., 1986. Auditory Icons: Using Sound in Computer Interfaces. *Human-Computer Interaction*, **2**(2), pp. 167-177.
- MORLEY, S., PETRIE, H., O'NEILL, A.M. and MCNALLY, P., 1998. Auditory navigation in hyperspace: design and evaluation of a non-visual hypermedia system for blind users. *Proceedings of the third international ACM conference on Assistive technologies*, , pp. 100-107.
- MORLEY, S., PETRIE, H., O'NEILL, A. and MCNALLY, P., 1999. Auditory navigation in hyperspace: design and evaluation of a non-visual hypermedia system for blind users. *Behaviour and Information Technology*, **18**(1), pp. 18-26.
- MYNATT, E.D., 1997. Transforming Graphical Interfaces Into Auditory Interfaces for Blind Users. *Human-Computer Interaction*, **12**(1 & 2), pp. 7-45.
- MYNATT, E.D., 1994. Designing with auditory icons: how well do we identify auditory cues? *Conference on Human Factors in Computing Systems*, , pp. 269-270.
- MYNATT, E.D., BACK, M., WANT, R., BAER, M. and ELLIS, J.B., 1998. Designing audio aura. *Proceedings of the SIGCHI conference on Human factors in computing systems*, , pp. 566-573.
- NICOL, C., BREWSTER, S. and GRAY, P., 2004. DESIGNING SOUND: TOWARDS A SYSTEM FOR DESIGNING AUDIO INTERFACES USING TIMBRE SPACES. *Tenth International Conference on Auditory Display (ICAD 04), Sydney, Australia*.
- WALKER, B.N., NANCE, A. and LINDSAY, J., 2006. SPEARCONS: SPEECH-BASED EARCONS IMPROVE NAVIGATION PERFORMANCE IN AUDITORY MENUS. *Proceedings of ICAD*, **2006**.