THE 'AUDIOVIEW' – PROVIDING A GLANCE AT JAVA SOURCE CODE

J Louise Finlayson

Department of Computing Science
University of Aberdeen
Aberdeen
Scotland
lfinlays@csd.abdn.ac.uk

Chris Mellish

Department of Computing Science
University of Aberdeen
Aberdeen
Scotland
cmellish@csd.abdn.ac.uk

ABSTRACT

Having an overview of the structure of information has been shown to be necessary to effectively approach the reading of it. This paper describes how programming constructs can be represented using speech and non-speech audio to provide an important 'glance' at program source code prior to reading it. Three methods of representing program code are investigated, using pure speech, non-speech and a combination of speech and non-speech to determine the most effective method to convey this type of information.

On the basis of these results, this paper concludes that non-speech sounds are able to successfully convey information about program structure. However, significantly better results are achieved when using speech output, either alone or in combination with the non-speech audio, with a significantly lower mental workload. These results suggest that earcons and non-speech sounds be used as a supplement to speech representations, rather than as an alternative.

1. INTRODUCTION

1.1 Background

It has been repeatedly proven that it is helpful to provide the blind user with an overview of information to enable them to plan how to approach the task of reading it [1] [2]. A sighted user can simply glance at the information and immediately pick up on a range of visual cues that aid their understanding of its context, structure and complexity. This important initial overview of the information is currently not available to blind computer users. There are also many other users who are unable to use a conventional computer screen for a variety of reasons and this difficulty is not solely limited to the visually impaired. For instance, a programmer debugging his program remotely via a mobile telephone might find this useful to navigate around his code.

This area of research was investigated by Robert David Stevens in the development of his 'MathTalk' system [2]. MathTalk was designed to allow blind students easier access to algebraic formulae. In this system, he recognised the need to have a general idea as to the structure of an algebraic expression in order to effectively plan how to approach the reading of it. Stevens investigated the use of earcons[1] to convey this information.

[1] Abstract musical tones that can be used in structured combinations to create 'sound messages'.

Stevens reported encouraging results, which showed that these algebra earcons were able to successfully convey the structural complexity and type of an expression. He suggested that some of his ideas could be incorporated into a programming environment, to provide an essential glance at the structure of computer source code prior to reading.

The notion of using music to convey programming constructs has been investigated by Vickers and Alty [3] [4]. In their study, they examined using earcons to provide information about the execution of a program to aid debugging behaviour and reported successful experiments using earcons to represent pairs of programming construct.

Neither of these studies involved a comparison of the effectiveness of earcons with other possible methods of representation (such as speech). In Stevens' study, a speech representation was discounted as 'too long' to be used for an overview (although speech was used elsewhere in MathTalk), whilst Vickers and Alty intended their earcon system to be used not as a glance, but as a supplement to the program text.

Anecdotal evidence would seem to suggest that a combination of different modes of information would increase task performance. In practise, this has not been proven. One problem in looking at the coordination of different modalities is the determination of what modes can be considered *different*. Studies investigating combinations of text and graphics consider these to be classed as different modes of information although other researchers may group both under the heading 'visual information'. In the auditory domain, the classification of speech and non-speech sounds is equally contentious.

There is a large body of psychological evidence to suggest that speech and non-speech sounds are processed differently by the human brain. Non-speech sound, unlike speech, does not cause the *suffix effect* when added to the end of a list of numbers [5]. Studies of brain activity show that speech stimuli produce greater activation bilaterally in the mid-superior temporal gyrus and adjacent superior temporal sulcus than non-speech stimuli [6] [7]. Lastly, 'Pure Word Deaf' patients have difficulty perceiving speech but not music or environmental sounds [8] [9].

Accepting that speech and non-speech are in fact different modes of information, there is little decisive evidence to indicate whether combining them will enhance task performance. Advocates of non-speech sound suggest that adding earcons and auditory icons to auditory interfaces will improve performance and prove less distracting for the user than adding extra lexical cues. Opponents, such as Donal Fitzpatrick [10] disagree. In his TechRead system, Fitzpatrick rejected earcons and other forms of non-speech as 'too distracting' preferring instead to utilise solely lexical information.

An obvious parallel can be drawn with studies into combining text and graphics. Research in this area has shown that, far from increasing task performance, combinations of text and graphics can be detrimental to performance [11].

This paper aims to investigate alternate methods of representing sections of Java source code, and to determine whether combining speech with non-speech audio improves or worsens task performance.

1.2 Objectives

The objectives of the study are to:

- Design and develop alternate 'AudioViews' of computer source code using pure speech, pure non-speech and combinations of speech and non-speech audio.
- Compare the effectiveness and difficulty of each of these 'AudioViews'.
- Determine the most suitable method to convey a glance of the information.
- Investigate whether combining different modes of information has any effect on task performance.

2. METHODOLOGY

2.1 Participants

The subjects used in the experiment all had some basic experience with Java programming, as it was necessary that the participants be familiar with the different types of Java programming construct.

The subjects used in the experiment were all sighted. This was deemed to be appropriate as previous studies have indicated that there is no significant difference between the performances of congenitally blind[1] and blindfolded subjects on this type of activity [12] and also because the intended beneficiaries of this research are not limited to the visually impaired.

2.2 Design

The first of the test conditions was the pure speech representation. In this condition each sample of program source code was edited to contain only structural information about the various programming constructs. For example, the sample code shown in figure 2.2.1 was altered to the representation shown in figure 2.2.2.

The pure speech version of the AudioView was spoken using the JAWS for Windows screenreading software.

[1] Adventitiously blind subjects did show a slightly improved performance as compared to the congenitally blind and blindfolded test groups.

```
public void fillArray() {

int[] arrayOfInts = new int[10];
for (int i = 0; i <10;  i++) {
        int j = (i*4)+7;
        arrayOfInts[i] = j;
        if (j%2 == 0){
               System.out.print("Array position"
                                + i + " is even");
        }
        System.out.println();
}

}// end method fillArray
```

Figure 2.2.1 Original sample of source code

"One statement.
For.
Two statements.
If.
One statement.
End If.
One Statement.
End For."

Figure 2.2.2 Pure speech AudioView of original sample

The second condition was the pure non-speech representation of the source code. In this condition, earcons were used to represent each programming construct and were combined to show the structural information of an excerpt of source code. The earcons used to produce the non-speech AudioView were created using existing established construction guidelines [13] [14]:

- Musical timbres are used instead of simple tones.
- Instruments from different families are used to aid discrimination between earcons (e.g. piano and trumpet as opposed to trombone and trumpet).
- Rhythms are made as different as possible with different numbers of notes in each rhythm.
- A short pause is inserted between combined earcons to prevent them running together.
- Pitch changes are not used as a discriminating factor.

Java programming constructs can be classified hierarchically (see figure 2.2.3). This study considers three distinct classes: sequence, selection and iteration. These can be further subdivided (e.g. iteration can be divided into FOR clauses and WHILE clauses). The hierarchical nature of the information greatly simplifies the process of earcon construction.

Using standard earcon construction guidelines, each basic class of construct (sequence, selection or iteration) is allocated a particular rhythm (shown in figure 2.2.4).

At the next level, the type of construct ('If' vs. 'Switch'; 'For' vs. 'While') is distinguished by adding a drumbeat to the 'If' and the 'While' statements. At the lowest level, discrimination between individual statements within a certain type ('If' vs. 'Else'; 'Do' vs. 'While' etc.) is achieved by altering the timbre of the earcon. 'If', 'Switch', 'Do' and 'For' statements are played in a piano sound whilst 'Else', 'Case' and 'While' statements are played using a marimba. The simple sequential

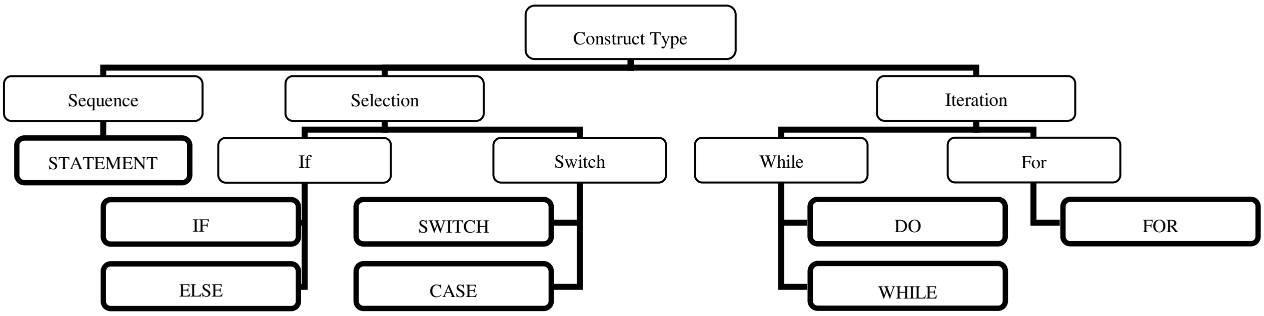
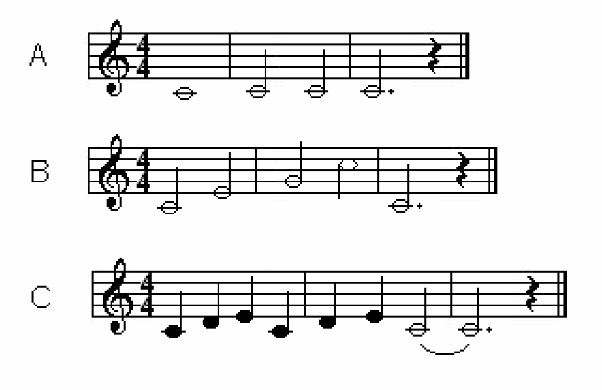


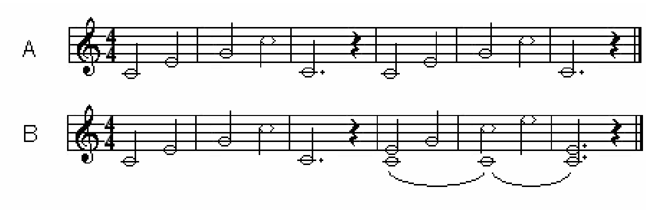
Figure 2.2.3 Basic Java Constructs



A: Sequence B: Selection C: Iteration

Figure 2.2.4 The three basic earcon rhythms

'Statement' is also played using piano and the amplitude of the sound is in direct proportion to the number of statements in the sequence. To represent construct nesting, the pitch of the earcon is increased at successive points in the major triad with each level of nesting and a drone is added using the previous notes of the chord. This is illustrated in figure 2.2.5.



A: Two selection statements- sequential (e.g. If{…}Else{…})
B: Two selection statement - nested (e.g. If{ If{…} })

Figure 2.2.5 A Non-speech AudioView showing construct nesting

The earcons were designed so that their length was comparable to the length of the speech representation to ensure a fair comparison between methods. The lengths of all types of view are in direct proportion to the length of the source code they represent so a longer code excerpt will always produce a longer view, independently of the method of presentation.

Preliminary investigation into the use of these earcons provided evidence that they could be used effectively via an experiment similar to that performed by Vickers and Alty [3] [4]. In the Vickers and Alty study, the subjects were presented with earcons representing pairs of nested or sequential constructs. The subjects were allowed to listen to each earcon three times and then were asked to identify the two construct types, and to state whether the second construct was sequential or nested. The earcons used in this study were evaluated in a similar manner (although subjects were only permitted to listen to each earcon twice), attaining a 93% accuracy rate compared with the 49% achieved by Vickers and Alty.

As with the Vickers and Alty experiments, the musical ability of the subjects was evaluated. This was assessed using a simple questionnaire (figure 2.2.6) to obtain an estimate of the subjects' musical aptitude on a scale ranging from zero to five, according to how many of the questions were answered positively. On this occasion there did seem to be a slight positive correlation between musical ability and performance however, the small sample number (only six) falls well short of the recommended 100 participants, which are required for an accurate evaluation of correlation.

1. Do you enjoy listening to music?

2. Have you ever or do you currently play any musical instrument (including voice)?

3. Have you ever or do you currently receive formal instrumental (or voice) tuition?

4. Do you have any formal musical qualifications (e.g. have you sat any music related examinations, either practical or theoretical)?

5. Can you read sheet music?

Figure 2.2.6 Musical Ability Questionnaire

The final test condition was a combination of speech and non-speech audio. In this condition, each sample of program source code was edited to contain only the structural information as in the pure speech example. However, the nesting levels were not indicated by speech but by playing successive notes from the C major tonic triad during the spoken rendition (i.e. one level of nesting is indicated by the note C, two levels are indicated by the notes C and E, and so on). Unlike in the pure speech condition, the exit from a nesting level was not indicated by a speech cue but simply by the removal of the relevant notes of the chord. The combination AudioView of the original sample of code shown in figure 2.2.1 is displayed in figure 2.2.7.

	"One statement.
	For.
C	*Two statements.*
C	*If.*
CE	*One statement.*
C	One Statement."

Figure 2.2.7 Combination speech and non-speech AudioView

This combination version of the AudioView, like the speech version, was spoken using JAWS for Windows. The accompanying chords were added using the JAWS' 'PlaySound' scripting function to play previously created wave files.

Sound samples for all three conditions can be found at "http://www.csd.abdn.ac.uk/~lfinlays/Projects/AudioView".

2.3 The Experiment

Training

For the pure speech and combination AudioViews the subjects were simply given a brief explanation of their construction before listening to up to eight typical examples.

For the pure non-speech group the training was more involved as it was necessary for the participants to first learn the relationship between the individual earcon and its corresponding programming construct. The subjects were given a detailed description of the construction rules and were allowed to listen to the earcons for each programming construct. Once familiar with the individual sounds, the subjects were presented with 28 sample AudioViews of nested and sequential construct pairs. Once familiar with these, they were provided with four longer examples, typical of those they would encounter in the testing phase of the experiment. The subjects were allowed an unlimited time to complete the training.

Procedure

Each subject took part in three separate experimental sessions, held at least a day apart - each investigating a different type of AudioView. A repeated measures design (i.e. each participant performs under all conditions of the experiment) was chosen as the best option as each subject acts as his own control. To counter the possibility of *order effect*s, the order in which each type of AudioView was presented was randomly allocated to each subject, with each ordering happening the same number of times. This was deemed an appropriate measure as it was assumed that any order effects would be symmetrical.

In each session, after training, the subject was presented with 8 AudioViews in total (each comprising a minimum of four constructs and possibly multiple nesting levels). The subject was presented with each AudioView twice, and then asked to identify the corresponding excerpt of source code from a choice of four examples. The 'wrong' examples were constructed so as to exemplify one of six typical errors: 1) omission; 2) addition; 3) alteration (same construct type); 4) alteration (different construct type); 5) inversion; and 6) nesting errors. The same test examples were used for each view. Figure 2.3.1 shows an example of the excerpts of source code used in the multiple choice questionnaire.

A
```
int[] arrayOfInts = {2,87,3,589,12,107,2,8,622};
for (int i = arrayOfInts.length; --i >= 0; ){
    for (int j = 0; j < i; j++){
        if (arrayOfInts[j] > arrayOfInts[j+1]){
            int temp = arrayOfInts[j];
             arrayOfInts[j] = arrayOfInts[j+1];
             arrayOfInts[j+1] = temp;
        }
    }
    for (int i = 0; i < arrayOfInts.length; i++){
        System.out.print(arrayOfInts[i] + " ");
    }
}
System.out.println();
```

B
```
int[] arrayOfInts = {2,87,3,589,12,107,2,8,622};
for (int i = arrayOfInts.length; --i >= 0; ){
    for (int j = 0; j < i; j++){
        if (arrayOfInts[j] > arrayOfInts[j+1]){
            int temp = arrayOfInts[j];
            arrayOfInts[j] = arrayOfInts[j+1];
            arrayOfInts[j+1] = temp;
        }
    }
}
for (int i = 0; i < arrayOfInts.length; i++){
    System.out.print(arrayOfInts[i] + " ");
}
```

C
```
int[] arrayOfInts = {2,87,3,589,12,107,2,8,622};
for (int i = arrayOfInts.length; --i >= 0; ){
    for (int j = 0; j < i; j++){
        if (arrayOfInts[j] > arrayOfInts[j+1]){
            int temp = arrayOfInts[j];
            arrayOfInts[j] = arrayOfInts[j+1];
            arrayOfInts[j+1] = temp;
        }
    }
}
for (int i = 0; i < arrayOfInts.length; i++){
    System.out.print(arrayOfInts[i] + " ");
}
System.out.println();
```

D
```
int[] arrayOfInts = {2,87,3,589,12,107,2,8,622};
for (int i = arrayOfInts.length; --i >= 0; ){
    for (int j = 0; j < i; j++){
        if (arrayOfInts[j] > arrayOfInts[j+1]){
            int temp = arrayOfInts[j];
            arrayOfInts[j] = arrayOfInts[j+1];
            arrayOfInts[j+1] = temp;
        }
    }
}
System.out.println("Results");
for (int i = 0; i < arrayOfInts.length; i++){
    System.out.print(arrayOfInts[i] + " ");
}
```

Figure 2.3.1 Sample of Multiple Choice Examples

In figure 2.3.1, option C is the correct answer. Option A shows a nesting error, option B highlights an omission error, while option D displays an error of inversion.

At the end of each session, the subject filled out a NASA TLX mental workload evaluation form to determine the relative difficulty of the AudioView. In this appraisal, subjects were asked to rate the difficulty of each type of AudioView in terms of physical demand, mental demand, temporal demand, effort, performance and frustration level. Each of these measures was then given a weighting according to its presumed relevance to the task. This was used to create the overall mental workload score ranging from 0 to 100 to reflect how demanding a subject determined each particular method to be. Finally, the subjects were asked to provide any additional comments that they thought might be of relevance.

3. RESULTS

The subjects' results under each of the three conditions are shown in table 3.1. The table shows the subjects' test scores for each condition and the nature of any errors that were made. The TLX mental workload scores for each type of AudioView are also shown.

These results were analysed using two-tailed paired t-tests to compare the test performance and mental workload involved with each condition. Table 3.2 shows the results of these analyses. p values deemed to be significant ($p < 0.05$) are marked with an asterix (*).

NON_SPEECH

Subject	score	percentage correct	error 1 (omission)	error 2 (addition)	error 3 (alt. sc)	error 4 (alt. dc)	error 5 (invers.)	error 6 (nesting)	mental workload
1	7	87.5%	0	0	0	0	0	1	68.8
2	5	62.5%	1	0	0	0	1	1	71.3
3	5	62.5%	1	1	1	0	0	2	73.0
4	5	62.5%	1	0	0	0	0	2	86.0
5	8	100.0%	0	0	0	0	0	0	79.9
6	7	87.5%	0	0	0	0	0	1	69.3
Mean	**6.2**	**77.1%**	**0.50**	**0.17**	**0.17**	**0.00**	**0.17**	**1.17**	**74.7**
St Dev	**1.3**	**16.6%**	**0.55**	**0.41**	**0.41**	**0.00**	**0.41**	**0.75**	**6.8**

COMBINATION

Subject	score	percentage correct	error 1 (omission)	error 2 (addition)	error 3 (alt. sc)	error 4 (alt. dc)	error 5 (invers.)	error 6 (nesting)	mental workload
1	8	100.0%	0	0	0	0	0	0	33.0
2	8	100.0%	0	0	0	0	0	0	33.0
3	8	100.0%	0	0	0	0	0	0	56.3
4	7	87.5%	0	0	0	0	0	1	42.3
5	8	100.0%	0	0	0	0	0	0	75.5
6	8	100.0%	0	0	0	0	0	0	57.3
Mean	**7.8**	**97.9%**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.17**	**49.6**
St Dev	**0.4**	**5.1%**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.41**	**16.6**

SPEECH

Subject	score	percentage correct	error 1 (omission)	error 2 (addition)	error 3 (alt. sc)	error 4 (alt. dc)	error 5 (invers.)	error 6 (nesting)	mental workload
1	8	100.0%	0	0	0	0	0	0	57.5
2	8	100.0%	0	0	0	0	0	0	20.2
3	6	75.0%	0	0	0	0	1	1	44.7
4	8	100.0%	0	0	0	0	0	0	48.6
5	8	100.0%	0	0	0	0	0	0	36.5
6	8	100.0%	0	0	0	0	0	0	56.5
Mean	**7.7**	**95.8%**	**0.00**	**0.00**	**0.00**	**0.00**	**0.17**	**0.17**	**44.0**
St Dev	**0.8**	**10.2%**	**0.00**	**0.00**	**0.00**	**0.00**	**0.41**	**0.41**	**14.0**

Table 3.1 Results of the subjects for each of the three conditions

	~~Speech v. Non-speech~~	~~Combination v. Non-speech~~	~~Speech v. Combination~~
Task performance	p = 0.030* t = -4.62 d = 5	p = 0.020* t = 3.37 d = 5	p = 0.070 t = -0.42 d = 5
Mental Workload	p = 0.006* t = 3.00 d = 5	p = 0.012* t = -3.81 d = 5	p = 0.551 t = -0.63 d = 5

Table 3.2 Paired t test results

4. DISCUSSION

Although the number of subjects was small, the results show that the subjects' performance for both the pure speech and the combination conditions was significantly better than in the non-speech condition. The TLX scores also show that the non-speech condition is significantly more mentally challenging.

The preliminary 'Vickers and Alty style' experiment produced a 93% identification rate of the construct pairs, comparing favourably to the 49% achieved in their previous study. This shows that the poor performance of the non-speech condition is not a result of poorly constructed earcons.

Unlike in the early earcon experiments, musical ability did not seem to be correlated with performance in the final non-speech AudioView condition. The subject with the lowest score on the musical aptitude scale held the lowest score on the preliminary earcon experiments yet scored the highest for the more complicated non-speech AudioView. This result may suggest that once familiar with the material, musical ability is no longer an important issue, and that sufficient training can overcome this factor.

The subjects' main comments on the non-speech condition were the speed at which the earcons were presented and the difficulties in discriminating between the different timbres. The speed issue was resolved with training, as once familiar with the material, the subjects found the speed of presentation less intimidating. The difficulties encountered in discriminating timbres was supported by the error analysis which showed that in the preliminary experiments 32% of errors were in identifying a particular sub-class of construct (i.e. 'if' vs. 'else') which were differentiated by a change in instrument. However this factor may also be overcome with training, as just 8% of errors in the final non-speech condition were due to timbre errors.

There was no significant difference between the pure speech and the combination conditions for either performance level or mental workload. This would seem to suggest that coordination of modalities, at least in this instance, was not detrimental to task performance. Subjects' cited preferences for each of these two conditions were varied, with different conditions being quoted as 'easier' by different subjects. No subjects indicated that the non-speech condition was easiest, however, they did state that it would probably get easier with practise.

The similarity between the performances of the combination and pure speech conditions may be because most of the excerpts of source code used in this study were fairly short and simple. Because of this, even the pure speech AudioViews could be held in short term memory (following Miller's 'seven plus or minus two' rule). Using longer code excerpts may result in a significant difference in the performances of the combined and pure speech conditions as combining speech and non-speech information increases the total amount of information that can be retained.

The results of this study indicate that to summarise language-type information, speech sound may be preferable to non-speech. Non-speech sound may prove to be more useful when summarising numerical or abstract information. The combined condition yielded slightly better results than the pure speech condition (although this difference did not reach significance). This may be evidence of the usefulness of non-speech in representing numerical variables (in this case, nesting levels).

This work provides evidence that earcons may be more successful if used in addition to speech or text rather than as a replacement. Although the performance of the non-speech AudioView was less successful than the other two methods investigated, the subjects performed much better than would have occurred by chance. Subjects remarked that they 'enjoyed' the non-speech condition as it was more pleasant to listen to and it may be a welcome addition to simple text or speech representations.

5. CONCLUSION

In conclusion, this study shows that earcons can indeed be used to represent programming constructs with relative accuracy. However, when compared to alternative methods of representation, such as speech, they are shown to be significantly less accurate and significantly more mentally challenging. This paper recommends that, to represent forms of language, earcons and non-speech sounds be used as a supplement to speech representations, rather than as an alternative. The indications are that combining modalities in this way will not be of detriment to performance.

This early experiment showed a slight (but not significant) tendency towards better results with the combination representation than in the pure speech condition. The results of the mental workload evaluation, however, revealed the speech representation to be regarded as slightly easier (also not significant). The next phase of this investigation will use longer, more realistic, examples to hopefully differentiate significantly between the effectiveness and relative difficulty of the pure speech and combination conditions.

6. ACKNOWLEDGEMENTS

This work was undertaken as part of a project being supported by a studentship awarded by the University of Aberdeen. We thank the reviewers for their comments and suggestions.

7. REFERENCES

[1] S. Morley, H. Petrie, A.M. O'Neill and P. McNally, "Auditory navigation in hyperspace: design and evaluation of a non-visual hypermedia system for blind users", *Behaviour and Information Technology,* vol. 18, no. 1, pp. 18-26, 1999.

[2] R.D. Stevens, "Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People", *PhD thesis*, University of York, 1996

[3] P. Vickers and J.L. Alty, "Using music to communicate computing information", *Interacting with Computers*, vol. 14, pp. 435-456, 2002.

[4] P. Vickers and J.L. Alty, "Musical program auralisation: a structured approach to motif design", *Interacting with Computers,* vol. 14, pp. 457-485, 2002.

[5] R.G. Crowder and J. Morton, "Precategorical acoustic storage (PAS)", *Perception and Psychophysics,* vol. 5, pp. 365-373, 1969.

[6] J.R. Binder., J.A. Frost, T.A. Hammeke, P.S.F. Bellgowan and J.A. Springer, "Human temporal lobe activation by speech and nonspeech sounds", *Cerebral Cortex,* vol. 10, pp. 512-528, 2000.

[7] D. Howard, K. Patterson, R. Wise, W.D. Brown and K. Friston, "The cortical localization of the lexicons", *Brain*, vol. 115, pp. 1769-1782, 1992.

[8] T. D. Griffiths, A. Rees and G.G.R. Green, "Disorders of human complex sound processing", *Neurocase*, vol. 5, pp. 365-378, 1999.

[9] D. Poeppel, "Pure word deafness and the bilateral processing of the speech code", *Cognitive Science*, vol. 25, pp. 679-693, 2001.

[10] D. Fitzpatrick, "Towards Accessible Technical Documents: Production of Speech and Braille Output from Formatted Documents", *PhD thesis*, Dublin City University, 2000.

[11] P. Romero, R. Cox, B. du Boulay and R. Lutz, "A survey of external representations employed in Object-Oriented programming environments", *Journal of Visual Languages and Computing*, vol. 14, pp. 387--419, 2003.

[12] L.H.D. Pol, "Visualising Graphical User Interfaces for Blind Users", *PhD thesis*, Techniche Universiteit Eindhoven, 1996.

[13] M. Blattner, D. Sumikawa and R. Greenberg, "Earcons and icons: Their structure and common design principles", *Human Computer Interaction*, vol. 4 no. 1, pp. 11-44, 1989.

[14] S.A. Brewster, P.C. Wright and A.D.N. Edwards, "A detailed investigation into the effectiveness of earcons". *Auditory display, sonification, audification and auditory interfaces. The Proceedings of the First International Conference on Auditory Display*, G. Kramer (Ed.), Santa Fe Institute, Santa Fe, NM: Addison-Wesley, pp. 471-498, 1992.